

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Réalisation d'un environnement de création de scénarios hypermédia dans le cadre du projet SYRECOS

Philippe, Yann

Award date:
1996

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique
rue Grandgagnage, 21, 5000Namur*

**Réalisation d'un environnement
de création de scénarios hypermédia
dans le cadre du projet SYRECOS**

Yann PHILIPPE

Promoteur : Monsieur le Professeur François BODART

Mémoire présenté en vue de l'obtention du grade de
Maître en Informatique

Année Académique 1995-1996

ABSTRACT

The project SYRECOS has for objective to create a multimedia interactive system of trades of competence as support to the regional development. It is in this framework that this thesis enters.

Its goal is to define a framework to develop a tool allowing the generation of scenarios hypermedia with the help of generic scenario models. These scenarios have to allow the utilization of the different possibilities provided by the hypermédia as well as containing multimedia forms. The technology WEB and its system hypermedia based on the language HTML represents the technological environment of the project.

This thesis proposes also some improvement perspectives of tools as well as of methodological perspectives. A brief criticizes of the RMM and Trident methods is proposed in the framework of their application to the hypermedia application.

RÉSUMÉ

Le projet SYRECOS a pour objectif de créer un réseau interactif multimédia d'échanges de compétences comme support au développement régional. C'est dans ce cadre que cette thèse s'inscrit.

Son but est de développer un outil permettant la génération de scénarios hypermédia à l'aide de modèles de scénarios génériques. Ces scénarios doivent permettre l'utilisation des différentes possibilités offertes par le multimédia ainsi que contenir des formulaires multimédias. La technologie WEB et son système hypermédia basé sur le langage HTML représente l'environnement technologique du projet.

Cette thèse propose aussi certaines perspectives d'amélioration des outils ainsi que des perspectives méthodologiques. Une brève critique des méthodes RMM et Trident est proposée dans le cadre de leur application à la conception d'application hypermédia.

REMERCIEMENTS

C'est pour moi un plaisir d'exprimer ma plus vive reconnaissance à toutes les personnes qui ont collaboré de près ou de loin, à la réalisation de ce mémoire.

Je tiens à remercier plus particulièrement :

Monsieur le Professeur François BODART, promoteur de ce mémoire, pour l'aide, les encouragements et les précieux conseils qu'il m'a prodigués tout au long de l'élaboration de ce travail.

Messieurs William POOS, Jean VANDERDONCKT et Jean-Marie LEHEUREUX pour leurs conseils et leurs idées, éclairant ainsi la réalisation de ce mémoire.

Messieurs Bryan HARP, Curt POWLEY, David BENJAMIN et toute l'équipe du projet DASHER pour leur accueil chaleureux et amical durant ce formidable stage à l'Information Science Institute (ISI) à Marina Del Rey (Los Angeles, USA).

Je remercie Valérie LEMAIRE pour son aide et son soutien lors de la rédaction de ce mémoire.

Je tiens également à remercier mes parents qui m'ont permis d'entreprendre ces années d'études et d'aboutir ainsi à la réalisation de ce mémoire.

Merci à tous.

TABLE DES MATIÈRES

CHAPITRE 1	1
INTRODUCTION GENERALE	1
I. INTRODUCTION	2
II. SYRECOS	4
CHAPITRE 2	7
HYPERTEXTE ET HYPERMEDIA	7
I. INTRODUCTION	8
II. DEFINITION	9
A. Généralités	9
B. Définition plus précise d'hypertexte	11
C. Hypermédia : hypertexte et multimédia	12
D. Hypertexte et les applications informatiques traditionnelles	14
E. L'enthousiasme à propos de l'hypertexte	15
III. HISTORIQUE	17
A. MEMEX (1945)	18
B. AUGMENT / LNS (1962-1976)	18
C. XANADU (1965)	19
D. HYPERTEXT EDITING SYSTEM (1967) and FRESS (1968)	19
E. ASPEN MOVIE MAP (1978)	20
F. KMS (1983)	20
G. HYPERTIES (1983)	21
H. NOTECARDS (1985)	21
I. INTERMEDIA (1985)	21
J. GUIDE (1986)	22
K. L'hypertexte s'accroît	22
IV. DES APPLICATIONS DE L'HYPERTEXTE	24
A. Applications informatique	24
B. Les applications dans le monde du business	25
1. Les manuels	25
2. L'audit	26
3. La loi	26
4. Les catalogues de produits, les publicités, ...	27
C. Applications dans le monde intellectuel :	27
1. Support à la réflexion et à l'organisation d'idées	27
2. Le journalisme	27
3. La recherche	28
D. Applications dans le monde de l'éducation	28
1. Les langues étrangères	28
2. Culture	28

3. Les musées	28
E. Les applications de spectacle et de loisirs	29
1. Les guides touristiques	29
2. Les librairies	29
3. La fiction interactive	29
4. Les journaux et magazines	30
5. Sexe	30

CHAPITRE 3 **33**

LE MONDE WEB **33**

I. INTRODUCTION	34
II. INTERNET	35
A. Historique	35
B. Le fonctionnement de l'INTERNET	36
C. Quelques grandes applications sur INTERNET	38
D. INTRANET	39
III. LE WORLD WIDE WEB	41
A. Historique	41
B. Le fonctionnement du WEB	41
IV. HYPERTEXT TRANSFER PROTOCOL	44
A. HTTP/9.0	44
B. HTTP/1.0	44
V. LE LANGAGE HTML	46
A. Structure générale d'un document HTML	46
B. Les en-têtes de documents	48
1. Le titre d'un document	48
2. La base d'un document	48
C. Le corps de documents HTML	48
1. Les titres de sections, les paragraphes et les tabulations	49
2. La mise en forme des caractères	49
3. Les liens vers d'autres documents	50
4. Les images	50
5. Les listes	51
6. Divers	51
VI. LE LANGAGE HTML+	52
A. Les formulaires	52
1. Déclaration d'un formulaire	53
2. Les champs d'édition	54
3. Les boîtes à cocher et les boutons radio	55
4. Les boîtes de sélection	55
5. Les boutons de commande	56
VII. LE COMMON GATEWAY INTERFACE (CGI)	57
A. Le lien entre les clients, le CGI et le serveur	57
B. Les variables d'environnement	59

CHAPITRE 4 **61**

LANGAGE DE DESCRIPTION DE SCENARIOS POUR SYRE COS **61**

I. INTRODUCTION	62
II. SCHEMA CONCEPTUEL DE BASE	65
A. Objets informationnels	65
1. Texte	65

2. Icône	68
3. Image	68
4. Son	69
5. Vidéo	70
6. Liste	70
7. Tableau	71
B. Objet interactif	72
1. EditText	72
2. MultiEditText	73
3. ListBox	74
4. ComboBox	74
5. ToggleButton	75
C. Section	76
D. Formulaire	76
E. Submit	76
F. Reset	77
G. Page	77
H. FrameSet	78
I. Lien	79
J. Scénario	80
K. Conclusion	80
II. ROLE DU SCENARISTE	81
III. EXEMPLE DE SCENARIO GENERIQUE	86
IV. ROLE DU REALISATEUR	101
A. Instantiation des scénarios	102
B. Instantiation des pages	102
C. Instantiation des objets informationnels	103
D. Instantiation des objets interactifs	103
E. Instantiation des liens	103
V. OUTILS	104
A. Outil de typage	104
1. Page Présentation	105
2. Page Couverture	106
B. Outil de NAVIGATION	106
1. Cadre conceptuel	106
2. Navigation séquentielle	108
3. Navigation indexée	109
4. Navigation guidée	110
5. Navigation mixte	111
6. Navigation globale	111
7. Exemple	112

CHAPITRE 5 **115**

REALISATION **115**

I. INTRODUCTION	116
II. L'ENVIRONNEMENT	117
A. le réseau capillaire namurois	118
III. IMPLEMENTATION	120
A. FRONTPAGE	121
1. Présentation	121
2. Utilisation par le scénariste	123
3. Utilisation par le réalisateur	124
B. SCENEWEB	124

1. Présentation	124
2. Utilisation par le scénariste	125
3. Perspectives et améliorations futures	132
IV. CAS PRATIQUE	133
CHAPITRE 6	139
PERSPECTIVES METHODOLOGIQUES ET CONCLUSION	139
I. PERSPECTIVES METHODOLOGIQUES	140
A. Relationship Management Methodology (RMM)	140
1. Relationship Management Data Model (RMDM)	140
2. Relationship Management Methodologie (RMM)	142
3. Conclusion	145
B. Méthodologie TRIDENT	145
1. La formalisation des spécifications IHM à partir de l'analyse de la tâche	146
2. Le pilotage actif de la présentation à partir de règles ergonomiques	149
3. La dérivation de l'architecture de l'application interactive à partir de l'analyse de la tâche et des composants de la présentation	150
4. La spécification d'un modèle de la conversation de haut niveau à partir de l'analyse de la tâche.	151
5. La particularisation du cadre de travail méthodologique en un cadre de travail de spécification	152
6. Conclusion	152
C. Perspectives d'une méthodologie dédiée à la conception d'applications hypermédia	152
II. CONCLUSION	154
BIBLIOGRAPHIE	155
ANNEXES 1	160
GRAMMAIRE BNF DU LANGAGE DE DESCRIPTION	160
ANNEXE 2	164
GRAMMAIRE BNF DE L'OUTIL DE TYPAGE	164
ANNEXE 3	165
GRAMMAIRE BNF DE L'OUTIL DE NAVIGATION	165
ANNEXE 4	166
SCHEMA PHYSIQUE DE LA DB POUR LA NAVIGATION	166
ANNEXE 5	167
GENERATION SQL DE LA DB POUR LA NAVIGATION	167

CHAPITRE 1

INTRODUCTION GÉNÉRALE

Ce chapitre est consacré à l'introduction et à la présentation du projet SYRECOS qui sert de cadre de travail à ce mémoire.

I. Introduction

Ce mémoire a pour objectif de réaliser un environnement de développement d'application hypermédia dans le cadre du projet SYRECOS. *« Le but du projet SYRECOS est de concevoir et réaliser un télé-service régional dédié à l'échange de compétences. Il privilégie d'une part à la formation professionnelle continue, et d'autre part au transfert de technologies »*¹. C'est dans le cadre de formation professionnelle continue que s'inscrit ce mémoire.

Après avoir présenté le projet SYRECOS plus en détail dans ce premier chapitre, nous tenterons de présenter un aperçu le plus large possible de ce qu'on entend par un système hypermédia. Quel pourrait être la définition de ce nouveau concept ? Est-ce réellement un nouveau concept informatique, quel en est son histoire ? Quels sont les domaines d'application des systèmes hypermédia ?

Le chapitre 3 fournit une introduction à Internet, à son histoire, à son fonctionnement et à ces outils et tout particulièrement le World Wide Web. Qu'est-ce que le World Wide Web, quel est son histoire, son fonctionnement ? Nous présenterons aussi le protocole HTTP, les langages HTML 3.0 et HTML+ et le Common Gateway Interface.

Dans le quatrième chapitre, nous définirons le cadre général de l'environnement de développement dédié au projet SYRECOS. Quel est le schéma conceptuel du langage de description nécessaire à la réalisation du système hypermédia dédié à SYRECOS ? Quels sont les objets dont nous avons besoin, quels outils peuvent nous aider dans la réalisation de scénario hypermédia ?

Le cinquième chapitre est consacré à la réalisation et à la mise en oeuvre de l'environnement tel que définit dans le chapitre 4. On présentera l'implémentation et l'utilisation des outils formant notre environnement de développement.

Finalement, le chapitre six présente une perspective méthodologique pour développer des applications hypermédia. Nous comparerons deux méthodes de conception et tenterons d'en dégager les grands avantages de l'une par rapport à l'autre pour donner les

¹ [HTTP ://WWW.INFO.FUNDP.AC.BE/~FTA/SYRECOS.HTML](http://WWW.INFO.FUNDP.AC.BE/~FTA/SYRECOS.HTML)

esquisses d'une méthode de conception d'application hypermédia. Ce chapitre clôturera ainsi ce travail après avoir présenté toutes les facettes du projet.

II. SYRECOS²

SYRECOS a vu le jour dans un soucis de redynamisation de notre économie. Cette redynamisation passe par une amélioration des conditions de développement des PME et PMI. Car c'est à ce niveau que l'on rencontre le plus de dynamisme, mais ces PME/PMI rencontrent des difficultés dues à leur petite taille. Cet intérêt au PME/PMI s'explique par le fait qu'elles composent notre tissu économique régional.

SYRECOS a donc pour objectif de soutenir et promouvoir les PME/PMI à l'aide des NTIC³. Car ces NTIC ont ouvert de nouvelles perspectives en matière commerciale et marketing grâce au développement de ce que l'on appelle les marchés électroniques.

SYRECOS s'est fixé comme objectif de prouver l'intérêt des NTIC, notamment par la mise en oeuvre de deux projets pilotes :

1. La formation continue
2. Le transfert des technologies

C'est dans le cadre de la formation continue que se place ce travail.

L'objet de SYRECOS est de développer un réseau interactif multimédia permettant les échanges de compétences et de services entre les différents acteurs régionaux (PME/PMI, Administration, citoyens, ...).

Le développement des télé-services proposé par SYRECOS propose un ensemble de fonctionnalités couvrant tous les aspects du cycle de vie de la formation continue. Nous renvoyons le lecteur désirant avoir plus d'information sur le cycle de vie d'une formation continue à [CIGER info, n°16, 1996, page 8].

Des fonctionnalités sont offertes aux organismes de formation qui désirent présenter leur organisme, présenter leur catalogue et leur calendrier de formation, les conditions de participation, ... Une autre série de fonctionnalités est offerte aux demandeurs de

² SYstème Régional d'Echange de Compétence et de Services

³ Nouvelles Technologie de l'Information et des Communications

formations qui peuvent accéder aux présentations des organismes de formation, à leur catalogue, s'inscrire à une formation, ...

C'est pour aider et faciliter la cohérence et la mise à disposition de ces informations sur les formations que nous développerons un environnement de création de scénario hypermédia. Le projet doit contribuer à la création rapide, par des non-informaticiens, d'applications de télé-commerce basées sur l'utilisation de systèmes hypermédia.

De manière générale, les projets hypermédia diffèrent des projets de développement traditionnels selon différents critères. Premièrement, les auteurs d'applications hypermédia doivent faire preuve de différentes qualités telles que celle d'auteur, concepteur, artiste, ergonomiste, scénariste, musicien, documentalistes et programmeur. Deuxième, la conception d'applications hypermédia implique la capture et l'organisation de structures d'informations complexes pour les rendre claires et accessibles aux utilisateurs. Troisièmement, les aspects hypermédia de l'hypermédia apportent un certain nombre de difficultés. Et finalement, le besoin de prototypage et de test est bien plus important ici que pour le développement d'applications traditionnelles, surtout en ce qui concerne le test de la navigation.

Ce sont ces raisons qui forment une barrière pour les petites PME/PMI. Nous proposons dans ce projet de réduire cette hétérogénéité en créant un environnement de développement de scénarios hypermédia basé sur la définition d'un modèle générique de scénarios de présentation d'activités, de produits et de services. La réalisation d'un scénario particulier par des non-informaticiens sera l'instanciation d'un scénario générique.

CHAPITRE 2

HYPERTEXTE ET HYPERMÉDIA

Ce chapitre est consacré à la définition des termes Hypertexte et Hypermédia, ainsi qu'à leur historique, leur évolution et une description des principaux domaines d'application de ces systèmes.

I. Introduction

Les systèmes hypertexte deviennent actuellement une nouvelle classe de systèmes complexes pour la gestion d'informations. Ces systèmes permettent aux personnes de créer, d'annoter, de lier et de partager des informations provenant d'une variété de médias différents tels que le texte, le graphique, l'audio, la vidéo, l'animation et les programmes.

Les systèmes d'hypertexte fournissent une méthode entièrement nouvelle et non séquentielle pour accéder à l'information, au contraire des systèmes d'information traditionnels qui sont séquentiels par nature. Ils fournissent des accès flexibles à l'information en incorporant les notions de navigation, d'annotation et de présentation adaptée.

Dans ce chapitre, nous allons nous attarder à définir ce que l'on entend par les termes hypertexte et hypermédia. Nous présenterons l'historique et l'évolution de l'hypertexte, et enfin nous montrerons quelques grands domaines d'application de l'hypertexte.

II. Définition

A. Généralités

La manière la plus simple de définir l'hypertexte est certainement celle présentée par [Nielsen 1995] Il le compare avec les textes traditionnels tel ce document. Tous les textes traditionnels qu'ils soient sous formes imprimées ou de fichiers, sont séquentiels. Cela signifie qu'il n'y a qu'une seule séquence linéaire définissant l'ordre dans lequel le texte doit être lu. Premièrement, vous lisez la page une, ensuite vous lisez la page deux, ensuite la page trois, et ainsi de suite.

L'hypertexte est d'une toute autre nature, il est non séquentiel. Il n'y a pas une seule manière qui détermine la séquence dans laquelle le texte va être lu.

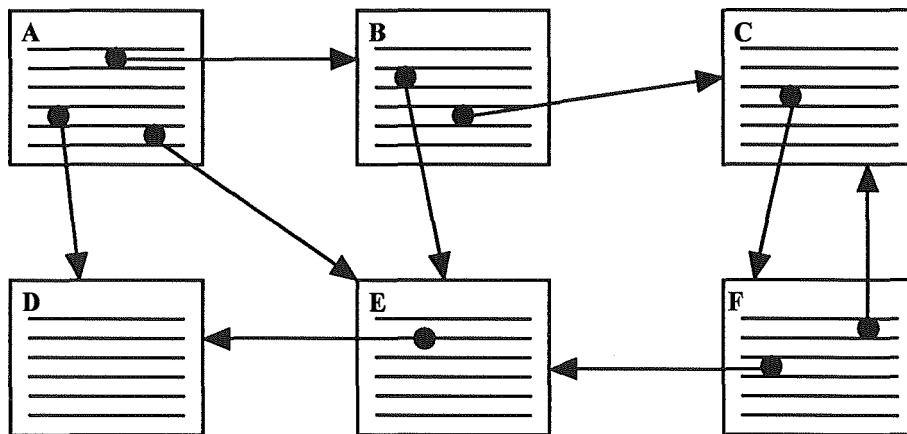


Figure 2.1: exemple de structure hypertexte

Prenons l'exemple illustré par la Figure 2.1. Supposons que l'on commence par lire la partie d'un texte appelée A, au lieu d'avoir une seule destination possible, cette structure d'hypertexte a trois options pour le lecteur : aller vers B, D ou E. Supposons qu'il décide d'aller vers B, il peut ensuite décider d'aller vers C ou vers E, et d'E il peut aller vers D. Bien qu'il ait été possible d'aller immédiatement d'A vers D. Cet exemple montre qu'il pourrait y avoir plusieurs chemins qui relient deux éléments d'une structure hypertexte.

L'hypertexte présente plusieurs options différentes pour les lecteurs et chaque lecteur détermine laquelle de celles-ci il décide de suivre au moment de la lecture du texte. Cela signifie que l'auteur du texte a défini un nombre d'alternatives pour que les lecteurs puissent l'explorer, plutôt que de suivre un chemin d'information unique.

La même chose est vraie pour les notes en bas de page dans les formats traditionnels de texte imprimé, puisque les lecteurs doivent déterminer s'ils désirent atteindre la note en bas de page plutôt que de continuer de lire le texte principal. L'encyclopédie avec ses multiples références croisées, est une autre forme imprimée qui a des structures d'accès similaires à l'hypertexte.

Comme le montre la Figure 2.1, l'hypertexte consiste en parties de texte ou d'autres informations, inter-reliées. Chaque unité d'information est appelée "noeud". Quelle que soit la granularité de ces noeuds, chacun d'eux peut avoir des pointeurs vers d'autres unités, et ces pointeurs sont appelés "lien". Le nombre de liens n'est normalement pas fixé à l'avance mais dépendra du contenu de chaque noeud. Certains noeuds sont liés à plusieurs autres et auront donc plusieurs liens, alors que d'autres noeuds peuvent juste servir de destination pour des liens, mais ne comprennent pas de lien sortant. Ce type de noeud est alors appelé '*feuille*' et est illustré dans la Figure 2.1 par la page D.

La Figure 2.1 montre aussi que la structure entière d'un hypertexte forme un réseau de noeuds et de liens. Pour bien mettre en évidence que les lecteurs doivent déterminer activement l'ordre dans lequel ils lisent les noeuds, on dit que les utilisateurs naviguent (Browse) dans la structure hypertexte.

Un lien hypertexte connecte deux noeuds et est normalement dirigé dans le sens où il pointe, à partir d'un noeud (appelé le noeud ancre) vers un autre noeud (appelé le noeud destination). Les liens hypertextes sont fréquemment associés à des parties spécifiques du noeud qu'ils connectent plutôt qu'avec le noeud entier. Dans l'exemple de la Figure 2.1, l'ancrage des liens se fait à une localisation spécifique dans le noeud de départ alors que leur destination est le noeud d'arrivée entier. Une application typique de cela est d'avoir un lien ancré dans un certain mot du noeud de départ et ensuite de laisser l'utilisateur activer le lien en cliquant sur ce mot (comme c'est le cas pour la majorité des liens du WEB).

Lorsque les utilisateurs suivent les liens dans le réseau hypertexte, ils auront souvent besoin de retourner au noeud précédemment visité. La plupart des systèmes hypertextes supportent cette facilité de "backtracking". Supposons que nous sommes actuellement positionné dans le noeud **D** de la Figure 2.1, si nous étions arrivé à ce noeud via le chemin **A** suivi de **B**, **E**, **D**; alors la commande de "backtracking" nous conduirait au

noeud **E**. Une seconde commande de backtracking pourrait ensuite nous amener plus loin dans notre chemin vers le noeud **B**. Par contre, si nous étions immédiatement passés d'**A** au noeud **D**, l'exécution du "backtracking" à partir du noeud **D**, nous emmènerait au noeud **A**. Cet exemple montre que le "backtracking" est dépendant des mouvements du lecteur.

B. Définition plus précise d'hypertexte

Etant donné que la popularité de l'hypertexte augmente ces dernières années, beaucoup de personnes essayent de donner une définition exacte de ce que constitue un hypertexte. Le terme '*hypertexte*' a été défini comme étant "*an approach to information management in which data is stored in a network of nodes connected by links. Nodes can contain text, graphics, audio, video, as well as source code or other forms of data*". [Smith and Weiss, 1988]

A cette définition de base, de nombreux auteurs ajoutent certaines caractéristiques supplémentaires qu'un système hypertexte devrait rencontrer. Par exemple, Franck HALASZ [Halasz, 1988] a avancé l'idée qu'un vrai système d'hypertexte devrait inclure une représentation explicite de la structure du réseau dans son interface utilisateur. Comme le montre la Figure 2.1, tous les hypertextes forment un réseau de noeuds et de liens, mais dans la plupart des systèmes actuels, ce réseau est seulement présent à l'intérieur de l'ordinateur. A tout moment, l'utilisateur ne voit que le noeud actuel et les liens permettant de se rendre vers d'autres noeuds à partir de celui-ci. La manière dont le réseau est structuré étant laissée à l'imagination de l'utilisateur.

HALASZ désire donner une vue dynamique à l'utilisateur lui montrant la structure de ce réseau. Actuellement, très peu de systèmes hypertexte fournissent de tels diagrammes⁴. La raison pour laquelle le diagramme doit être dynamique est qu'il est normalement impossible de dessiner une représentation graphique de l'hypertexte en entier sur l'écran d'un ordinateur, étant donné qu'un hypertexte contient typiquement des milliers de noeuds. Par contre, le diagramme est affiché en détail, c'est-à-dire uniquement pour les voisins locaux autour de la localisation actuelle de l'utilisateur.

Pratiquement tous les systèmes hypertextes actuels ne fournissent que des liens unidirectionnels tels que ceux de la Figure 2.1. Cela signifie que le système peut montrer

⁴Le système de Halasz, NoteCards, est l'une de ces exceptions.

à l'utilisateur les liens que comprend le noeud actuel comme point de départ, mais pas ceux qui ont ce noeud comme point d'arrivée. En d'autres termes, le système dira où on peut aller mais ne dira pas quelles sont les différentes alternatives qui auraient pu nous faire arriver à ce noeud.

K. Eric Drexler a défendu l'utilisation de liens bidirectionnels dans les hypertextes. Cela signifie que le système devrait aussi être capable d'afficher la liste des liens entrants. D'un point de vue informatique, cela serait pratiquement une tâche triviale à implémenter, mais la plupart des systèmes hypertexte actuels ne le fournissent pas.

Drexler a aussi relevé la nécessité de supporter des liens à travers différentes formes de réseaux informatiques tels que : LAN et International network. Cette étape deviendra nécessaire si l'hypertexte est appelé à remplacer le monde traditionnel de la publication, puisque personne ne peut avoir toute la littérature du monde stockée sur son propre ordinateur local et ce, quel que soit la taille des disques optiques dont il dispose. L'accès à des bases de données distantes deviendra une nécessité pour beaucoup d'applications hypertexte futures, mais beaucoup de systèmes hypertexte actuels sont limités à travailler avec des données stockées sur un ordinateur personnel unique. Les principales exceptions sont le World Wide Web, qui travaille à travers le réseau Internet entier, et les systèmes de documents techniques tels que SUN's ANSWERBOOK qui travaille à travers un réseau privé. Nous reviendrons sur le WWW dans un chapitre suivant.

D'après Jakob Nielsen [Nielsen,1995], beaucoup de techniques informatiques non hypertexte peuvent rencontrer au moins certains des critères de la définition d'hypertexte, mais les vrais systèmes hypertexte devraient aussi faire sentir à l'utilisateur qu'il peut naviguer librement à travers les informations, selon ses propres besoins. Ce sentiment est difficile à définir précisément mais implique certainement une facilité d'utilisation. Cela signifie aussi un temps de réponse court de telle sorte que le texte soit affiché à l'écran dès que l'utilisateur l'a demandé. Cela implique également peu de charge cognitive lors de la navigation, de telle sorte que l'utilisateur n'ait pas à perdre son temps pour savoir ce que l'ordinateur va faire, ou comment arriver à ce qu'il fasse ce que l'on désire.

C. Hypermédia : hypertexte et multimédia

La définition traditionnelle du terme hypertexte implique un système traitant du texte. Depuis que beaucoup de systèmes actuels incluent aussi la possibilité de travailler avec

des graphiques et différents autres médias, certaines personnes préfèrent utiliser le terme d'hypermédia pour gérer les aspects multimédia du système. Dans ce document, on utilisera aussi bien le terme d'hypertexte que celui d'hypermédia pour décrire les mêmes systèmes.

En tout cas, l'hypertexte est une technique naturelle pour supporter des interfaces multimédias puisqu'il est basé sur l'interconnexion de noeuds qui peuvent contenir différents médias. Les médias typiques d'un noeud hypermédia sont le texte, le graphique, la vidéo ou le son.

Les graphiques peuvent être aussi bien des photos scannées ou des images construites par certaines applications informatiques. Ils peuvent être la forme principale de l'information dans un système. Ils peuvent être utilisés simplement comme illustration dans un système où les liens sont restreints au texte, ou ils peuvent être plus activement impliqué dans les aspects hypertexte d'un système hypermédia en incluant l'ancrage de lien hypertexte.

Différentes images animées formant une vidéo ou animation sont aussi des types de données communs dans les noeuds hypermédia.

Une difficulté avec la représentation vidéo dans l'hypertexte est de savoir comment nommer les liens. Les solutions les plus traditionnelles ont été d'utiliser du texte comme point d'ancrage hypertexte menant à l'exécution d'une partie de la vidéo, mais ce n'est pas un choix très orienté hypermédia.

Le son dans le système hypertexte introduit encore un autre problème de lien. Il est très facile d'avoir un son comme destination pour un lien hypertexte; le son est exécuté quand l'ancre est activée. Mais dans beaucoup d'application on pourrait désirer que le point d'ancrage de départ soit le son lui-même. Si l'utilisateur désire naviguer dans la musique, la conception de l'interface devient plus problématique, il serait très difficile pour l'utilisateur de mentionner une partie spécifique de musique de manière directe. Finalement, il est tout simplement impossible de cliquer sur un son, de la même manière que l'on puisse cliquer sur un mot ou une image. Au lieu de cela, il faudrait fournir à l'utilisateur une représentation visuelle quelconque du son, sur laquelle il puisse cliquer. Par exemple dans le cas de la musique, une représentation pourrait être les notes de

musique ou les textes des paroles, pour d'autres sons une représentation moins intuitive devrait être utilisée.

D. Hypertexte et les applications informatiques traditionnelles

L'hypertexte est fondamentalement différent des bases de données traditionnelles d'un point de vue de l'utilisateur. Une base de données normale telle que la base de données des employés d'une compagnie, a une structure extrêmement régulière, définie par un langage de définition de données de haut niveau. Toutes les données suivent cette structure unique, donc on pourrait avoir 10000 enregistrements d'employés ayant tous les mêmes champs pour le nom, l'adresse, le salaire, le numéro de téléphone, etc. ...

Il est vrai que le système hypertexte a beaucoup de similarités avec les bases de données, étant donné qu'on a besoin d'une forme de base de données en dessous du système hypertexte pour stocker et retrouver le texte et les autres médias contenus dans les noeuds.

Une base d'information hypertexte n'a pas de définition centrale et de structure régulière, on peut encore avoir 10000 noeuds dans le cas d'un système hypertexte pour les employés d'une compagnie. Mais maintenant certains de ces noeuds sont très extensibles avec beaucoup d'informations, et d'autres sont relativement pauvres. Certains employés peuvent travailler sur des projets avec des collègues d'autres divisions, et pourraient donc contenir des liens à partir de la distribution de leur travail vers les noeuds de ces autres employés. En général, la structure d'un réseau hypertexte est définie comme l'union de décisions locales faites dans la construction de chacun des noeuds et des liens individuels. Chaque lien y est mis, car il a du sens en terme de contenu sémantique des deux noeuds qu'il connecte et non pas à cause de certaines décisions globales. Cela signifie qu'un hypertexte est fortement flexible, ce qui est normalement un avantage mais peut aussi être un désavantage.

Les jeux d'aventure sont des classes d'applications informatiques qui sont basées sur la navigation. Ils peuvent être vu comme un système hypertexte d'après certaines définitions et même, dans certains cas, des informaticiens ont implémentés des jeux d'aventure sur base d'un système hypertexte. Mais d'après Jakob Nielsen [Nielsen,1995], de telles applications ne sont pas des systèmes d'hypertexte car ils sont fondamentalement basés

sur le fait de rendre la navigation difficile à l'utilisateur pour arriver à la destination désirée, et souvent ces applications cachent les liens vers d'autres localisations. Ce qui est un contraste complet avec les contraintes cognitives pour l'interface utilisateur d'un système hypertexte.

Comme il a déjà été mentionné, bien que beaucoup de systèmes hypertexte soient aussi des systèmes hypermédia et incluent des effets multimédia, le fait qu'un système soit orienté multimédia ne le rend pas nécessairement hypertexte. Le mélange de textes et de graphiques n'est pas assez en soi. Beaucoup de systèmes multimédia sont orientés principalement sur l'affichage de différents clips vidéo pour un utilisateur passif qui n'a pas à naviguer dans un espace d'informations. C'est seulement quand l'utilisateur prend contrôle de manière interactive d'un ensemble de liens dynamiques parmi les unités d'informations, que l'on peut qualifier ce système d'hypertexte. Par analogie on peut illustrer cette différence entre le multimédia et l'hypermédia, par la différence entre regarder un film de voyage et être un touriste soit même.

Un type de système multimédia qui est souvent confondu avec l'hypermédia est la vidéo interactive. Encore une fois, il est certainement possible d'utiliser les effets de la vidéo interactive dans une interface hypermédia, mais beaucoup de ces systèmes vidéo interactifs ne sont pas suffisamment interactifs pour être classé comme hypermédia. Le caractère principal ici est la manière de laquelle l'utilisateur a la possibilité de déterminer les activités du système. Beaucoup de systèmes de vidéo interactives réduisent l'utilisateur au rôle d'un visionneur de télévision passif qui a seulement la possibilité de sélectionner les clips vidéo à partir d'un menu. La raison pour laquelle on ne peut parler d'hypertexte dans ce cas, est que l'utilisateur n'a aucune possibilité d'interagir avec le clip vidéo une fois qu'il est lancé. En d'autres mots, l'interaction est trop faible pour fournir à l'utilisateur le sentiment de contrôler et la capacité d'explorer l'espace d'informations.

E. L'enthousiasme à propos de l'hypertexte

L'un des avantages les plus importants de l'hypertexte est qu'il s'agit d'une méthode permettant d'intégrer trois technologies et industries qui ont été séparées jusqu'à maintenant : la publication (création, édition, imprimerie), l'informatique et la diffusion (distribution). Il y a eu quelques essais de méthodes non hypertexte pour réunir ces trois industries, telle que la publication de cassettes vidéo ou l'utilisation de l'informatique dans les départements de l'éditorial des journaux. Mais en général chacune de ces trois

industries continua de travailler comme avant jusqu'à cette convergence récente qui a mené à la fusion de plusieurs compagnies de ces trois types d'industries.

L'hypertexte fournit l'opportunité de publier des structures d'informations pour le grand public de la même manière que les livres ou les journaux sont actuellement publiés. Ces structures d'information seraient basées principalement sur des images animées dans la tradition de l'industrie du film et de l'animation, et seraient sous contrôle informatique pour permettre l'interaction de l'utilisateur.

III. Historique

Cette section retrace l'évolution des grands systèmes hypertexte, des grandes idées, implémentées ou non. D'après [Nielsen,1995], le Tableau 2.1 illustre les événements marquants de l'histoire des systèmes hypertexte.

1945	Vannevar Bush proposes Memex
1965	Ted Nelson coins the word 'hypertext'
1967	The Hypertext Editing System and FRESS, Brown University, Andy van Dam
1968	Doug Engelbart demo of NLS system at FJCC
1975	ZOG (now KMS) : CMU
1978	Aspen Movie Map, first hypermedia videodisk, Andy Lippman, MIT Architecture Machine Group
1984	Filevision from Telos ; limited hypermedia database widely available for the Macintosh
1985	Symbolic Document Examiner, Janet Walker
1985	Intermedia, Brown university, Norman Meyrowitz
1986	OWL introduces Guide, first widely available hypertext
1987	Apple introduces HyperCard, Bill Atkinson
1987	Hypertext'87 Workshop, north Carolina
1991	World Wide Web at CERN becomes first global hypertext, Tim Berners-Lee
1992	New-York Times Book Review cover story on hypertext fiction
1993	Mosaic anointed Internet killer app, National Center for Supercomputing Applications
1993	A Hard Day's Night becomes the first full-length feature film in hypermedia

1993	Hypermedia encyclopedias sell more copies than print encyclopedias
------	--

Tableau 2.1: événements principaux de l'histoire de l'hypertexte

A. MEMEX (1945)

L'idée originelle de l'hypertexte a été émise pour la première fois par Vannevar BUSH en juillet 1945 [Bush, 1945]. Il décrit un périphérique appelé MEMEX dans lequel un individu stocke ses livres, enregistrements et communications et ceux-ci sont mécanisés de telle sorte qu'ils peuvent être consultés de manière beaucoup plus rapide et flexible. Il s'agit d'un supplément personnel et élargi de sa mémoire. Vannevar BUSH décrit l'apport essentiel de MEMEX comme étant sa capacité à nouer deux éléments ensemble.

Bien que le système n'ait jamais été implémenté, BUSH est considéré comme le grand-père de l'hypertexte car ses concepts sont encore pertinents aujourd'hui. Le MEMEX (MEMory EXtender) stockerait ces informations sur des microfilms conservés dans le bureau de l'utilisateur. Ce bureau était destiné à avoir plusieurs écrans pour la projection des microfilms afin de permettre à l'utilisateur de comparer différents microfilms de la même manière qu'aujourd'hui on utilise plusieurs fenêtres sur notre PC. Il y aurait aussi un clavier et un ensemble de boutons et de leviers.

Le MEMEX aurait eu aussi un scanner pour permettre l'introduction de nouvelles matières par l'utilisateur. Et il aurait aussi permis aux utilisateurs de rajouter des notes et commentaires manuscrits et personnels. A part la forme traditionnelle d'un index, BUSH proposa : *'associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of MEMEX. The process of tying two things together is the important thing.'*

B. AUGMENT / LNS (1962-1976)

Douglas ENGELBART [Engelbart, 1963] commença à travailler en 1962 sur le projet AUGMENT qui avait pour but de développer des outils informatiques afin d'augmenter les capacités humaines et la productivité. Ce projet était le premier travail majeur dans les domaines tels que le traitement de texte et l'automation bureautique. Le projet a été conduit au SRI (Standfort Reserch Institute).

Une partie du projet AUGMENT était LNS (On Line System) qui avait plusieurs éléments hypertexte, bien qu'il n'ait pas été développé comme un système hypertexte. Ce système était utilisé pour stocker tous les papiers de recherche, les mémos, les rapports dans un espace de travail partagé qui pourrait être inter-référencé avec d'autres. En 1968 il fit une démonstration de LNS comme étant un système de collaboration entre des personnes géographiquement éloignées.

C. XANADU (1965)

Ted NELSON a pour la première fois utilisé le mot hypertexte en 1965. Il travailla sur sa vision de "docuverse" (DOCUMENT uniVERSE) où tout serait accessible à tout le monde. N'importe quel utilisateur serait capable de suivre des origines et des liens à travers les limites de documents, de serveurs, de réseaux et d'implémentations individuelles. Il y aurait un environnement unifié disponible pour tous et fournissant l'accès à cet espace complet [Nelson, 1987].

NELSON conçut XANADU, un système de répertoires de publication '*intended to store a body of writings as an interconnected whole, with linkages, and to provide instantaneous access to any writings within that body.*' [Nelson, 1980]. Ce système n'a pas de concept de suppression, c'est un système '*Write Ones*'. Une fois que quelque chose est publié, c'est pour que le monde entier puisse le voir à jamais. Comme les liens sont créés par les utilisateurs, le document original reste le même, à l'exception du fait qu'une nouvelle version est créée, laquelle référencerait la version originale. Étant donné que les systèmes de fichiers conventionnels ne sont pas adéquats pour implémenter un tel système hypermédia. Le projet XANADU a principalement centré son attention sur la conception et l'implémentation d'un tel système de fichiers. Ce qui entraîne aussi la création d'un nouveau système d'exploitation incorporant un moteur hypertexte. Ce projet commence seulement à être partiellement implémenté.

D. HYPERTEXT EDITING SYSTEM (1967) and FRESS (1968)

L'HYPertext EDITING SYSTEM construit en 1967 était le tout premier système hypertexte au monde fonctionnant. Il fut trouvé lors d'un contrat de recherche IBM. Après que L'HYPertext EDITING SYSTEM fut terminé comme projet de recherche à l'université de Brown, IBM le vendit à Houston Manned Spacecraft Center où il est

actuellement utilisé pour produire la documentation des missions APOLLO. Le second système hypertexte était FRESS (File Retrieval and Editing System) qui fut réalisé à l'université de Brown en 1968 comme la suite de l'HYPertext EDITING SYSTEM, et aussi implémenté sur les mainframes d'IBM.

Ces deux systèmes d'hypertexte avaient les fonctionnalités de base hypertexte de liens et de sauts vers d'autres documents. Mais la plupart des interfaces utilisateurs étaient sur une base textuelle et nécessitaient des spécifications indirectes de l'utilisateur pour les sauver.

E. ASPEN MOVIE MAP (1978)

Probablement le premier système hypermédia, développé par Andrew LIPPMAN et ses collègues du MIT. ASPEN était une application de transport qui permettait à l'utilisateur de conduire de manière simulée à travers la ville de ASPEN, sur un écran d'ordinateur. Le système ASPEN était implémenté avec un ensemble de vidéodisks contenant des photographies de toutes les rues de la cité de ASPEN Collorado. Des photos de toutes les rues de la ville étaient prises par quatre caméras orientées à 90° d'intervalle. L'aspect hypermédia du système vient de la manière d'accéder à ces images. Cela ne se faisait pas d'une manière traditionnelle comme des bases de données (Donnez-moi '150 Main street') mais comme un ensemble d'informations liées. Chaque photographie était liée aux autres. Une personne pouvait décider de continuer tout droit, retourner en arrière, tourner à gauche ou tourner à droite.

F. KMS (1983)

Le KNOWLEDGE MANAGEMENT SYSTEM (KMS) développé à Carnegie Mellon University a probablement la distinction d'être le plus vieux parmi les systèmes hypertexte actuellement populaires. Il fut conçu pour gérer de larges réseaux hypertexte à travers un réseau local (LAN). L'unité de base du système KMS s'appelle le frame. Le frame peut contenir du texte, du graphique ou de l'image. Les frames sont connectés à d'autres frames via les liens. Les liens sont de deux types : éléments d'un arbre pour représenter les relations hiérarchiques et éléments d'annotation pour représenter les relations de référence. Dans KMS il n'y a pas de distinction entre le mode de navigation et de création. Les utilisateurs peuvent faire des changements à un frame ou créer un lien à tout moment, ces changements sont automatiquement sauvegardés [Acksyn, 1988].

G. HYPERTIES (1983)

HYPERTIES fut au départ '*TIES, The Interactive Encyclopedia System*', sous la direction de Ben SHNEIDERMAN à l'université de Maryland. Il fournit des outils de création et de navigation. Un noeud peut contenir un article entier qui consiste en plusieurs pages. Les liens sont représentés par des mots mis en évidence ou des menus emboîtés qui peuvent être activés en utilisant le clavier ou le touchscreen. Les lecteurs peuvent avoir un aperçu des liens avant de les traverser. L'interface utilisateur est relativement simple étant donné que le contexte d'origine fut les systèmes d'informations des musées ou des kiosques. La version commerciale est actuellement utilisée dans un spectre d'applications bien plus large telle que la résolution de problèmes de diagnostic, l'aide on-line, les browser pour des librairies, ...[Schneiderman, 1988]

H. NOTECARDS (1985)

NOTECARDS est un système hypermédia pour les concepteurs, les auteurs et les chercheurs afin d'analyser l'information, construire des modèles, formuler des arguments et exécuter des idées [Halasz, 1988b]. Son cadre de travail de base est un réseau sémantique composé de carnets de notes connectés par des liens typés. Il fournit aux utilisateurs des outils pour l'affichage, la modification, la manipulation et la navigation à travers le réseau.

NOTECARDS contient quatre constructions de base : les carnets de notes, les liens, le browser et les boîtes de fichiers. NOTECARDS contient des informations imbriquées dans le texte, les graphiques, les images, la voix ou les autres médias. Les liens représentent les relations binaires entre les notes. Le browser affiche des portions du réseau par des diagrammes de noeuds et de liens. Les boîtes de fichiers fournissent un mécanisme pour organiser les notes en sujets ou catégories. NOTECARDS peut être intégré avec d'autres systèmes s'exécutant dans un environnement lisp tel que le système de courrier, les data bases et les systèmes experts.

I. INTERMEDIA (1985)

INTERMEDIA est un environnement hypertexte fortement intégré, développé à Brown University. Cet environnement intégré permet l'utilisation de différents types d'applications tels que les traitements de texte, les éditeurs ou d'autres programmes, et il peut les lier ensemble. C'est une collection d'outils qui permet aux auteurs de lier

ensemble le contenu de textes, de graphiques, de modèles 3D, de documents vidéo à travers un réseau de stations de travail performantes. [Meyrowitz, 1986]

Les applications qui existent dans une structure INTERMEDIA incluent un traitement de texte (InterText), un éditeur graphique (InterDraw), un visionneur d'images scannées (InterPix), un visionneur d'images 3d (InterSpect) et un éditeur de lignes du temps (InterVal).

La fonctionnalité hypermédia du système est intégrée dans chaque application de telle sorte que la création de liens peut être mixée avec la création et traversée de liens. Le système fournit une interface utilisateur standard à travers toutes les applications de telle sorte qu'il est facile pour l'utilisateur d'interagir avec toutes les applications de la même manière.

INTERMEDIA supporte le concept de WEB, les entités composites qui ont beaucoup de noeuds et de liens entre eux. Un lien peut appartenir à un ou plusieurs WEB. Il supporte les accès partagés des concurrents au document basé sur un système de permission d'accès. INTERMEDIA a été utilisé pour présenter deux cours en direct. [Yankelovich, 1988]

J. *GUIDE (1986)*

GUIDE a été développé par Peter BROWN dans un projet de recherche de l'Université de Kent in UK. C'est le système hypertexte commercial le plus populaire. Les textes et les graphiques sont intégrés ensemble dans des articles ou documents. GUIDE supporte quatre types de liens : des boutons de remplacement, des boutons de note, des boutons de référence et des boutons de commande. La navigation par les boutons de remplacement qui fournissent initialement un résumé de l'information et le degré de détail peut ensuite être changé par le lecteur. De la même manière que KMS, guide ne fait aucune distinction entre la création et la lecture.[Brown, 1987]

K. *L'hypertexte s'accroît*

Au milieu de l'année 1990, les systèmes hypermédia firent leur apparition pour le grand public à travers la prolifération de CD-ROM. On peut citer parmi ceux-ci le CD-ROM réalisé par la compagnie Voyager sur le film des Beatles "A hard days's night". Ce CD-ROM fut le premier à contenir l'intégrale du clip sous une forme hypermédia. Un autre

événement des années 1990 a été l'extraordinaire expansion de l'hypertexte sur Internet avec comme fer de lance : les spécifications du World Wide Web par Tim BERNARD-LEE et ses collègues du CERN (Centre Européen pour la Recherche sur la physique Nucléaire). Quasi immédiatement après l'introduction par le NCSA en Janvier 1993, Mosaïc devint l'outil de navigation le plus populaire pour le WWW. Et l'évolution de l'hypertexte sur Internet s'accéléra encore.

Le chapitre trois traitera plus en profondeur de l'hypertexte sur Internet. Mais, il est d'ores et déjà intéressant de remarquer le fait que Mosaïc et le WWW ont réussi à établir un système d'hypertexte universel en juste trois ans alors que Ted NELSON a eu besoin de trente ans d'essais pour faire accepter son système XANADU. La raison principale de cette différence est sans doute le fait que le projet WWW fut financé par les contribuables, ce qui rend plus facile de vendre un produit quand il coûte zéro francs. Bien sur, il y a aussi d'autres raisons pour lesquelles WWW a réussi et XANADU a échoué. La différence la plus importante est que le WWW est par nature, un système ouvert. Les concepteurs du WWW ont conçu leur système pour travailler avec l'Internet à travers des standards ouverts, avec la capacité de traiter le type des données qui furent disponibles sur le réseau à ce moment. Ces spécifications ont assuré le succès du WWW mais ont aussi limité sa capacité à fournir tous les moyens que l'on désirerait, de manière idéale, pour un système hypertexte.

IV. Des applications de l'hypertexte

On retrouve beaucoup d'applications basées sur l'hypertexte, elles peuvent être dans des domaines aussi variés que des applications informatiques, liées au business, basées sur des activités intellectuelles, pour les loisirs, pour l'éducation, ... Malgré cette diversité d'applications, il est à noter que toutes les applications ne devraient pas être réalisées en hypertexte. Pour déterminer si une application convient pour l'hypertexte, Ben SHNEIDERMAN a proposé ce qu'il appelle trois règles d'or pour l'hypertexte. Première : une large quantité d'informations est organisée en un nombre de fragments. Deuxième : les fragments se rapportent à chacun des autres. Troisième : l'utilisateur n'a besoin que d'une fraction des informations à chaque moment.

A. Applications informatique

Etant donné que l'hypertexte est un moyen informatique, il est naturel qu'il soit utilisé avec des applications plutôt orientées informatique.

Documentation on-line : il s'agit de l'application hypertexte la plus naturelle et la plus ancienne. Etant donné qu'aucun utilisateur ne lit complètement les manuels, il faut mettre à leur disposition des outils qui leur permettent d'avoir un accès facile à l'aide pour qu'ils puissent retrouver les sections nécessaires à leur besoin ponctuel. L'hypertexte est la méthode la plus évidente pour aider l'utilisateur dans une telle situation, d'ailleurs actuellement la plupart des applications informatiques contiennent des systèmes d'aide on-line. Un exemple de documentation on-line est le système d'aide de Microsoft Office 4.0.

Assistance à l'utilisateur : l'hypertexte fournit un mécanisme pour intégrer différentes formes d'assistance aux utilisateurs tel que l'assistance on-line, l'éditorial introductif, les systèmes d'aide on-line et même des erreurs messages. Dans un système qui fournit cette facilité, il devrait être possible pour un utilisateur de trouver, à partir d'un message d'erreur, une partie du système d'aide qui l'assistera à résoudre son problème. Il devrait pouvoir trouver de l'information à partir de chacun des mots du message d'erreur, et s'il désire avoir plus d'informations sur cette partie, le système devrait lui fournir un lien vers le chapitre approprié dans le système d'aide. De tels systèmes commencent à exister avec l'apparition de Windows 95 et son système de débogage et d'aide à l'utilisateur pour

l'installation et la paramétrisation de nouveaux périphériques. La station Sun 386i comprend est son système d'aide on-line pour l'assistance à l'utilisateur implémenté comme hypertexte [Campagnoni and Ehrlich 1989].

Ingénierie logiciel : pendant le cycle de vie de développement d'un logiciel, un grand nombre de documents, de spécifications et d'implémentations sont produits et l'hypertexte a un grand potentiel pour fournir des liens parmi ceux-ci. Par exemple, il devrait être possible à partir d'un besoin spécifié dans la cahier des charges de naviguer vers la partie de spécification qui rencontre ce besoin et il pourrait ensuite lier la spécification au code pour voir comment la conception de ce besoin a été implémentée. Il serait très utile aussi de faire la même chose dans le chemin inverse, c'est-à-dire à partir du code voir quel besoin cela rencontre. On peut citer Dynamic Design Project à Tektronix [Bigelow 1988] qui contient un support de contrôle des versions et Graphical Issue Based Information System (gIBIS) [Conklin and Begeman 1988] comme systèmes intéressants d'ingénierie.

- Les systèmes d'exploitation : l'hypertexte a le potentiel pour révolutionner l'interface utilisateur des ordinateurs en général, dans le but d'arriver à un environnement de travail pour des tâches intégrées. Habituellement les ordinateurs sont fondamentalement basé sur un paradigme de fichiers chaque fichier peut typiquement être trouvé dans une seule localisation du système de fichiers. Des liens hypertexte permettraient à l'utilisateur de passer d'un morceau d'information à un autre tout en lui évitant la tâche ardue de s'abandonner dans le système de fichiers. Les utilisateurs pourront se concentrer sur leur tâche et avoir un ordinateur intégrant ces applications et ces données pour rencontrer ces tâches.

B. Les applications dans le monde du business

1. Les manuels

Beaucoup de compagnies automobiles fournissent leurs instructions à leurs mécaniciens sur la manière de réparer les voitures sous une forme d'hypertexte, sur un support CD-ROM. Ils contiennent une description, enrichie grâce à la vidéo, sur la manière de réparer les voiture. Celles-ci expliquent au mécanicien comment réaliser les différentes opérations et montrent les erreurs typiques ainsi que ce qui pourrait mal se dérouler

pendant certaines opérations. L'hypertexte et le CD-ROM sont dans ce cas très avantageux, vu le nombre des différents modèles et manuels qu'une distribution complète nécessiterait. Beaucoup de dictionnaires et grands livres thématiques ont été convertis à partir d'un support papier traditionnel vers un format hypertexte, mais très peu ont réellement été généré pour un système hypertexte à partir de zéro. On commence seulement à trouver des dictionnaires français réalisés dans une philosophie hypertexte. La plupart des premières versions n'étaient qu'un dictionnaire auquel on avait rajouté quelques liens et un support multimédia tel que le son, la vidéo, ... Ce sont d'ailleurs ces médias qui fournissent le plus gros avantage aux encyclopédies et dictionnaires sous forme hypertexte comparé aux formes traditionnelles imprimées. L'Oxford English Dictionary. peut être cité comme exemple de dictionnaire converti en une version hypertexte.

2. L'audit

L'audit est une autre application naturelle de l'hypertexte car basé sur des informations reliées entre différentes sources et la vérification de leur consistance. La tâche de l'audit inclut le rassemblement et la production d'un grand nombre de documents et la liaison entre ceux-ci pour en extraire l'exactitude des informations qu'ils contiennent. Un grand nombre d'informations se retrouve réduit dans un simple rapport financier donc les liens entre la conclusion et les informations de source sont nécessaires. On peut retrouver cette application de l'hypertexte dans un prototype développé à Price Waterhouse Technology Center in Menlo Park [De Young 1989]

3. La loi

L'hypertexte a deux applications principales dans le domaine légale. Le premier est de supporter les recherches légales, ce qui est principalement utilisé par les juristes professionnels. La seconde est le support de la création de documents légaux et leur gestion, ce qui est principalement utilisé par des juristes professionnels et personnes qui veulent éviter de payer des amendes. Les textes légaux sont remplis de références croisées, ce qui correspond parfaitement à un support hypertexte. Par exemple, un avocat qui doit se référer à plusieurs décisions de tribunaux qui elles mêmes se basent sur plusieurs règles et ainsi de suite, le système hypertexte permet donc de faire gagner beaucoup de temps aux juristes. La compagnie de téléphone Nynex [France 1994]

illustre bien ce type d'application de l'hypertexte en utilisant un système de production de contrats.

4. Les catalogues de produits, les publicités, ...

Beaucoup de types de publicités, de communications à la clientèle peuvent être améliorés par l'hypertexte, ne serait-ce que par sa valeur en tant que nouveauté, qui peut être un avantage dans certains types de publicité. Par exemple, on peut tout simplement attirer l'attention à un certain stand commercial par le simple fait d'avoir un ordinateur avec l'information hypertexte sur nos produits. L'hypertexte pourrait aussi fournir de l'information à propos d'un palais des expositions complet et aider les gens à trouver les exposants qui les intéressent. Un catalogue de produits hypertexte peut réduire la complexité de choisir parmi un grand nombre d'options tout simplement en montrant celles qui peuvent être intéressantes pour le client. D'ailleurs ce type d'application commerciale se développe actuellement sur Internet.

C. Applications dans le monde intellectuel :

1. Support à la réflexion et à l'organisation d'idées

L'hypertexte est peut-être la manière la plus naturelle pour organiser des idées humaines car sa structure de réseau sémantique correspond au cerveau humain. Des systèmes hypertexte peuvent être utilisés pour organiser des discussions. Etant donné que l'hypertexte permet de coordonner différentes parties de texte, il peut aussi être utilisé pour organiser les idées d'un groupe de travail. L'hypertexte permet aussi à n'importe quel utilisateur de rajouter des notations, de rajouter des liens ou autres. On a déjà cité gIBIS comme outil d'organisation de discussion mais il y a aussi NOTECARDS de Xerox [Halasz, 1987], mais on peut aussi citer le système SEPIA [Streitz, 1992].

2. Le journalisme

En plus de la possibilité à long terme d'avoir des journaux dans un système hypermédia intégré, il est aussi possible d'utiliser l'hypertexte dès aujourd'hui dans le journalisme. Les mécanismes hypertexte peuvent très bien aider les journalistes à organiser leurs idées mais surtout peuvent les aider à rassembler, rechercher des informations parmi la très grande quantité d'informations qu'ils doivent traiter. Tous ces avantages ont été très bien perçus par The New York Times.

3. La recherche

La recherche est aussi un domaine où typiquement tous les documents référencent d'autres théories. Donc l'hypertexte peut très bien aider à ces références croisées et aider les chercheurs dans leur génération d'idées, et leur organisation. Par exemple on a écrit un document sur la biologie à l'aide de KMS [Schnase and Leggett 1989].

D. Applications dans le monde de l'éducation

Le système hypertexte est très intéressant pour les applications d'apprentissage où il est permis à l'étudiant une certaine liberté d'action et où on l'encourage à prendre des initiatives. Un des systèmes hypertexte d'éducation est le CD-ROM sur la découverte de Palenque [Wilson 1988]. Texas A&M University a aussi expérimenté un support hypertexte pour les aider les professeurs de cours universitaires [Legget, 1989].

1. Les langues étrangères

Les capacités de liens de l'hypertexte sont idéales pour l'apprentissage des langues étrangères. L'hypertexte peut fournir des accès automatiques au dictionnaire à partir de n'importe quel lien implicite d'un texte. L'hypertexte permet aussi aux étudiants de voir en parallèle deux versions du même texte, par exemple la version originale et la traduction. Evidemment l'hypermédia permet aussi de lier le son et l'image, ce qui permet de rendre les cours bien plus attractifs. 'A la rencontre de Philippe', du projet Athena du MIT [Hodges, 1989] a pour but de faire apprendre le français par un jeu de rôle.

2. Culture

La prolifération actuelle des CD-ROM nous montre que l'hypertexte est un système idéal pour apprendre notre histoire, nos cultures, nos artistes,... Il permet de regrouper différentes sources de données, aussi bien son, texte et vidéo par thèmes. Il permet de faire des rapprochements, des analyses de certains documents, certaines peintures, ... L'hypertexte fut utilisé comme support d'apprentissage de la littérature grecque, de l'histoire et de l'archéologie dans le projet 'The Perseus' de l'université d'Harvard [Crane, 1987],[Crane, 1988], [Marchionini and Crane, 1994] et [Mylonas, 1992].

3. Les musées

Etant donné que la plupart des gens vont de moins en moins au musée, les systèmes hypertexte sont un bon moyen intermédiaire pour tout de même diffuser notre culture. Il

est impossible de présenter les musées avec toutes les informations nécessaires et intéressantes à propos des expositions sous une forme imprimée. Les systèmes hypertexte nous permettent d'avoir accès à différents niveaux de connaissances selon qu'on soit un amateur ou un professionnel avec déjà certaines bases sur le sujet ou non. La difficulté de connaître l'étendue d'un espace d'informations hypertexte est actuellement un avantage pour ce type d'application. De plus les utilisateurs peuvent accéder à des informations de différentes manières. Le CD ROM qui permet de visiter le Louvre est un très bon exemple de cette application de l'hypertexte.

E. Les applications de spectacle et de loisirs

1. Les guides touristiques

Les informations touristiques rencontrent parfaitement les règles d'or de l'hypertexte, étant donné que les touristes ne désirent généralement avoir accès qu'à une petite partie de l'information disponible à propos d'une ville ou d'un pays donné. De plus l'information peut raisonnablement et facilement être divisée en noeuds pour chaque attraction ou service touristique, aire historique ou bien localisation géographique. Le problème avec les guides touristiques traditionnels est qu'ils ont besoin de structurer toutes ces informations selon un principe unique, alors que le touriste a de multiples besoins. Un des guides touristiques hypertexte est 'Glasgow Online' [Baird and Percival, 1989].

2. Les librairies

Quelques applications de librairie existent pour retrouver des informations techniques ou scientifiques. Elles sont très similaires aux applications telles que les dictionnaires ou les techniques de recherche d'information. Mais l'on pourrait imaginer que si les librairies veulent se moderniser et soient dans le futur non plus un bâtiment mais un réseau d'ordinateurs. Dans un tel contexte les techniques hypertexte pourraient aider les utilisateurs à trouver les livres sans les limites des techniques traditionnelles de recherche d'informations. Il pourrait permettre différentes stratégies de recherche pour trouver un livre. Exemple : le projet 'The Book House' [Pejtersen, 1989].

3. La fiction interactive

D'après Jakob NIELSEN [Nielsen, 1995], on aurait très peu gagné de la conversion de la fiction dans une forme traditionnelle, c'est-à-dire linéaire, en un système hypertexte hypermédia. Aussi longtemps qu'on lira un roman avec une seule suite d'action, on ferait

mieux de continuer à lire un livre imprimé. Par contre, si de nouvelles formes de fictions sont inventées on pourrait tirer profit à les convertir en hypertexte. Le lecteur a besoin d'interagir avec l'univers de la fiction plutôt que juste être un robot qui tourne les pages.

Une possibilité pour ce type de fiction serait un univers partagé. Ce type de fiction est récente mais très populaire dans le genre de science-fiction. L'idée de base est que plusieurs auteurs écrivent un ensemble d'histoires dans le même univers de fiction avec le même arrière plan et beaucoup de personnages communs. On collecterait plusieurs centaines de telles histoires dans un système hypertexte et on laisserait le lecteur poursuivre le type d'intrigue et les personnages qu'il trouve intéressant. Un tel projet de fiction interactive pourrait rencontrer les trois règles d'or car il y aurait plusieurs petits éléments qui peuvent être inter-reliés et être agréables pour le lecteur qui verrait juste une partie de ces éléments. Une des grandes fictions interactives pour enfants est le CR-ROM : 'The Manhole'.

4. Les journaux et magazines

La diffusion d'articles de presse est un candidat très intéressant pour les services interactifs étant donné que les événements peuvent être rapportés alors qu'ils se déroulent, au lieu d'avoir à attendre la période de publication des journaux et magazines. Grâce au système hypertexte on pourrait accéder à l'information qui nous intéresse par le biais de possibilités de recherche, trouver toute l'information sur un sujet des derniers mois, et même avoir accès à bien plus d'informations et éviter une étape de filtrage et d'interprétation. 'Interchange online service' procure l'accès aux articles de divers magazines informatisés [Perkins 1995].

5. Sexe

Etant donné la nature humaine, le fait que la plupart des nouveaux médias sont utilisés dans des buts sexuels, érotiques et pornographiques ne devraient pas être une surprise. Les cassettes vidéo, le téléphone, la photographie, les livres, la peinture, tous ces médias ont été utilisés pour la pornographie. Suivant cette ligne, le marché hypermédia a vu naître un grand nombre de produits de sexualité et pornographiques avec les CD-ROM interactifs et érotiques. A long terme, le fait que l'hypermédia est dynamique et est contrôlé par l'utilisateur, offrira un grand potentiel pour ces applications particulières. Il permet de concevoir plus d'expériences interactives que celles offertes par les médias statiques. Une preuve de la dominance du marché du sexe est démontrée par les

newsgroups sur Internet, où quatre des dix groupes les plus lus sont à connotation sexuelle. 'Virtual Valerie' est en haut de la liste des ventes des CD ROM érotiques.

CHAPITRE 3

LE MONDE WEB

Ce chapitre est consacré à la présentation du réseau INTERNET, de ses fondements, ses outils et tout particulièrement, le WEB. Il présente aussi le langage HTML, son extension HTML+ et l'utilisation de programmes via le Common Gateway Interface.

I. Introduction

Notre projet s'inscrit dans le contexte d'un INTRANET. La mise à disposition d'informations sera sous forme HTML et la consultation de ces informations se fera par l'entremise d'un browser standard. Dans ce chapitre, nous apporterons quelques précisions concernant tout ce qui touche à ce domaine. Dans un premier temps, nous expliquerons ce que désignent les termes INTERNET, WEB et le nouveau venu : INTRANET. Quels sont leurs origines, leurs fondements, leurs fonctionnements,...

Ensuite, nous présenterons d'une façon détaillée le langage HTML (HyperText Markup Language). C'est celui-ci qui est utilisé pour créer des pages dans le monde WEB. La dernière partie traite des CGI (Common Gateway Interface) qui sont les programmes exécutables sur un serveur. Ces programmes permettent au serveur de traiter des requêtes et sont tout particulièrement utilisés pour traiter les réponses à un formulaire.

II. INTERNET

Avant de voir plus en détail ce qu'est INTERNET [Krol and Hoffman, 1993], on peut déjà dire qu'il s'agit d'un énorme réseau mondiale. De manière plus précise, INTERNET est un réseau de réseaux, c'est-à-dire un ensemble de sous-réseaux répandus dans le monde entier et qui sont interconnectés ensemble. D'ailleurs, le terme INTERNET vient de *INTERconnected NETworks*. Ces sous-réseaux communiquent entre eux grâce à des protocoles de communication, c'est-à-dire grâce à un ensemble de conventions qui définissent comment les différents sous-réseaux peuvent s'échanger de l'information.

A. Historique

Le début d'INTERNET date de la fin des années 60, où l'ARPA (Advanced Research Project Agency), département de recherche de l'armée américaine, a mis au point un réseau, alors appelé ARPANET, qui avait pour but d'assurer un service de communication entre les différentes agences armées, les bases, ... et cela dans un contexte de guerre froide, où la menace nucléaire soviétique était d'actualité.

Avec le temps, de plus en plus d'universités et de centres de recherches utilisèrent aussi cet extraordinaire outil de communication. Au point où ARPANET leur a été entièrement abandonné et où l'on créa un nouveau réseau MILNET pour l'armée.

Parallèlement à l'évolution d'ARPANET, de plus en plus de réseaux virent le jour dans le monde entier, grâce à l'évolution des outils de communications. Il fallut donc trouver un moyen pour relier cet énorme potentiel de communication, trouver un protocole de communication permettant la connexion et la communication entre ces différents réseaux.

Et c'est lors de l'avènement de TCP/IP (Transmission Control Protocol / Internet Protocol) que les fondements de l'Internet tel que nous le connaissons aujourd'hui, furent établis.

C'est seulement vers le début des années 1990 qu'INTERNET s'ouvrit au domaine commerciale, avec une extraordinaire ascension du nombre de machines interconnectées et du nombre de sites.

B. Le fonctionnement de l'INTERNET

Le coeur de l'INTERNET est son protocole : TCP/IP. L'utilisation de TCP/IP permet une indépendance par rapport à la machine et au système d'exploitation. Ce sont des protocoles ouverts et distribués gratuitement. Enfin, ils sont basés sur schéma d'adressage commun. La pile TCP/IP peut se subdiviser en couches du monde OSI, comme le montre la Figure 3.2.

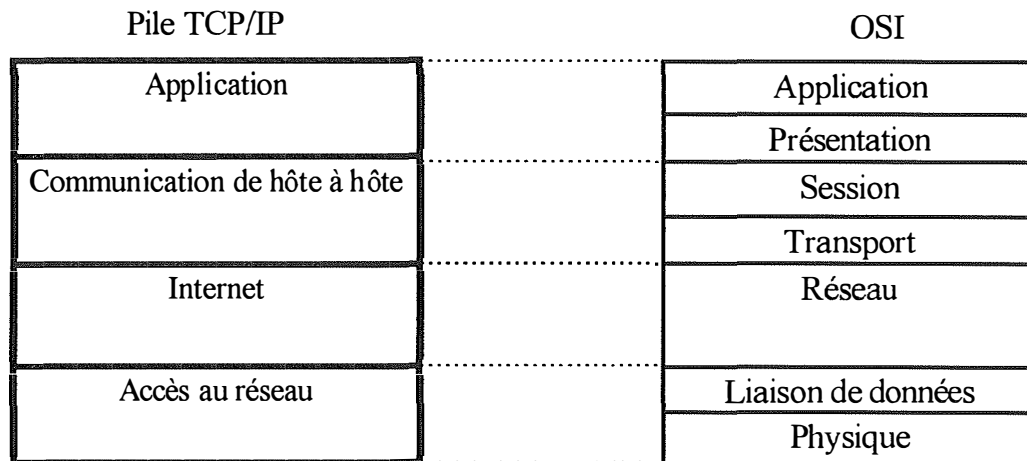


Figure 3.2: Comparaison de la découpe en couche des modèles TCP/IP et OSI

La couche d'accès au réseau donne au système les moyens de délivrer les données aux machines connectées localement. Cette couche reçoit de la couche Internet des datagrammes et les transforme en trames qui seront transmises sur le réseau. La deuxième fonction principale de cette couche est la conversion des adresses INTERNET en adresses physiques du réseau local.

Le protocole IP, compris dans la couche INTERNET, fournit les services d'acheminement des paquets. Pour ce faire, il gère la définition des datagrammes, le routage des datagrammes jusqu'à leur destination, la définition du schéma d'adressage, et enfin, la fragmentation et le réassemblage des datagrammes.

IP est un protocole connection-less; il garantit uniquement la transmission des données au réseau. Les datagrammes IP ont une entête qui contient les adresses de source et de destination, des informations quant à la fragmentation éventuelle des données et des informations à propos du protocole utilisé.

L'architecture de l'INTERNET est basée sur une collection de réseaux autonomes que l'on nomme les domaines de routage, il s'agit en fait des sous-réseaux composant INTERNET.

Chaque machine connectée sur INTERNET possède une adresse, appelée adresse IP qui l'identifie au sein d'un sous-réseau particulier. Une adresse de machine est donc unique et aucune autre machine ne pourra avoir la même au sein du même domaine. Certaines machines servant de passerelles entre plusieurs sous-réseaux ont plusieurs adresses IP (une par sous-réseau), mais en général, une machine a une seule adresse réseau. Les adresses sont actuellement codées sur 32 bits mais risquent de passer à 128 bits avec la nouvelle définition d'IP qui sera IPv6, étant donné la prolifération du nombre de machines à connecter. Les adresses IP sont composées de quatre numéros séparés par des points (ex :193.70.1.26). Les numéros de gauche, appelés bits de poids fort, permettent d'identifier le domaine et les bits de droite, les bits de poids faibles, désignent les machines dans le domaine, il s'agit des points d'accès. Dans notre exemple, 193.70.1 représente l'adresse du domaine et 26 l'adresse d'une machine dans ce sous-réseau.

On retrouve trois classes d'adresses IP, comme nous l'illustrons dans la Figure 3.3, où les cases grises représentent la partie domaine et la blanche la partie machine.

CLASSE A :	0-128	0-255	0-255	0-255
CLASSE B :	129-255	0-128	0-255	0-255
CLASSE C :	129-255	129-255	0-255	0-255

Figure 3.3: Classes d'adressage pour INTERNET

La gestion des adresses INTERNET est hiérarchique : le NICC (Network Information Control Center) distribue des fourchettes d'adresses à des organismes officiels qui se chargent d'allouer des adresses aux organismes demandeurs. La partie machine de l'adresse est gérée localement. Les trois types de formats d'adresses permettent une certaine souplesse vis-à-vis de la taille des réseaux physiques. Une adresse de classe A autorise 128 réseaux avec chacun 16 millions de points d'accès. Une adresse de classe B permet 16 mille réseaux et 64 mille machines. Enfin, la classe C assure 2 millions de réseaux et 256 machines.

Le problème de ces adresses est que la mémorisation de quatre chiffres n'est pas facile pour l'esprit humain. Une adresse plus parlante est plus facilement mémorisable. C'est

dans cet objectif d'utiliser des noms plutôt que des chiffres que l'on utilise un système attribuant des noms aux ordinateurs. Il s'agit du Domain Name Service (DNS). Le format de ces noms ressemble au format adopté par IP, il s'agit d'une suite de noms séparés par des points (par exemple : leibniz.info.fundp.ac.be). Contrairement à IP, la partie domaine se situe à droite alors que la machine (Leibniz) se situe à gauche. Le nom du domaine permet également de fournir une certaine information sur la machine qu'il référence. Dans notre exemple, *be* signifie que la machine est située Belgique, *ac* pour Academic, *fundp* pour l'université de Namur et *info* pour l'institut d'informatique.

Pour la couche de transport, deux protocoles principaux existent : TCP et UDP. TCP propose un service de bout en bout, avec détection et correction des erreurs. UDP assure un service par datagrammes, sans connexion. Les développeurs peuvent choisir quel type de couche de transport convient le mieux à leur application.

Le protocole TCP utilise le mécanisme de Positive Acknowledgement Retransmission. Cela signifie que chaque segment émis est acquitté par le destinataire. Et si l'émetteur ne reçoit pas d'accusé de réception après un certain délai, il émettra le segment à nouveau. Les entêtes des segments comprennent : les ports sources et destinations, le numéro de séquence et la taille de la fenêtre de réception.

C. Quelques grandes applications sur INTERNET

Le protocole IP est donc le protocole d'interconnexion de réseaux permettant à ces derniers de communiquer entre eux, mais, bien entendu, il faut des applications qui puissent utiliser ce protocole. Voici une liste des applications les plus courantes sur INTERNET :

- **FTP** : (File Transfer Protocol) est à la fois une application et le protocole sous-jacent à celle-ci. Le but du protocole FTP est d'assurer un standard pour le transfert de fichier d'une machine à une autre, permettant de récupérer des fichiers situés sur des machines éloignées.
- **Email** : (Electronic mail) Les applications de courrier électronique permettent d'envoyer des messages à un utilisateur défini sur une machine particulière et de garder ces messages à la disposition du destinataire, jusqu'à ce que celui-ci les

lisent. (principe de Store and forward) Le protocole sous-jacent à ces applications et le plus utilisé est SMTP (Simple Mail Transfer Protocol)

- **Telnet** : Telnet désigne à la fois une application et son protocole. Il permet à un utilisateur de transformer sa machine en un terminal d'un serveur éloigné, à condition de posséder un accès (login) sur ce serveur.
- **Newsgroups** : Cette application permet aux utilisateurs de communiquer entre eux à propos de certains sujets particuliers via un système de messages postés que tous les utilisateurs accédant au même newsgroup peuvent lire et y répondre.
- **WWW** : (World Wide Web) Il s'agit d'un système d'information hypermédia sur INTERNET. Mais cette application est plus détaillée dans la section qui suit.

Il existe plusieurs autres applications et protocoles sur INTERNET telles que GOPHER, WAIS, ... En théorie, il peut exister une infinité d'applications pour INTERNET puisqu'il suffit pour qu'une application communique via INTERNET qu'elle utilise le protocole TCP/IP.

D. INTRANET

On définit généralement un INTRANET en le limitant à un réseau d'entreprise (LAN) basé sur le protocole de communication TCP/IP et utilisant des services HTTP (services WEB), SMTP (services de messageries INTERNET), NNTP (Network News Transfer Protocol) et FTP (File Transfert protocol, gestion de fichiers et répertoires). En réalité, un réseau qualifié d'INTRANET peut être bien plus qu'un LAN, il peut s'agir de n'importe quel type de réseau (LAN, MAN, WAN, ...) basé sur la technologie IP. Il s'agit donc d'un MiddleWare.

Quasi natifs dans un environnement UNIX, ces éléments demandent des passerelles dans les environnements LAN. Ces passerelles sont désormais banalisées et pour la plupart livrées avec les OS. Microsoft inclut tous ces éléments dans ses OS : Windows 95 et NT. L'INTRANET se conforme aux standards publiés par les organismes normalisateurs de l'INTERNET et peut donc se muer facilement, en tout ou en partie, en site INTERNET.

L'avantage majeur de l'INTRANET est d'être accessible aux utilisateurs via une suite d'applications clientes standardisées, intuitives, légères et très peu coûteuses.

Finalement, on peut dire que l'INTRANET est un INTERNET limité à un réseau plus petit que le monde entier, ayant les mêmes capacités et les mêmes outils qu'INTERNET. D'ailleurs, dans ce document, à chaque fois que l'on parlera d'INTERNET, on pourrait transposer la phrase avec INTRANET.

III. Le World Wide Web

A. Historique

Le World Wide Web a été créé en mars 1989, par le professeur Tim BERNERS-LEE, du Centre Européen de Recherche Nucléaire (CERN) à Genève, lorsqu'il détailla la proposition de développer un système hypertexte. L'objectif premier du WEB était de faciliter la communication des informations et la publications des résultats scientifiques au sein du CERN. En effet, les chercheurs sont répartis dans de nombreux pays et il était donc intéressant de développer un produit leur permettant un échange précis et rapide.

Le projet de départ était de réaliser un logiciel permettant la navigation hypertexte à travers un réseau. Il n'était nullement question de gérer des documents sonores, vidéo ou des images. C'est seulement au début de l'année 1991 que la première démonstration de leur logiciel de navigation eut lieu à la conférence Hypertext 91. Depuis lors, l'engouement pour le WEB n'a cessé de croître, surtout avec l'introduction du multimédia dans le système hypertexte.

Depuis l'idée créatrice, le WEB a encore mûri. Le WEB est maintenant vu comme un vase de connaissances universelles. C'est un outil permettant la navigation hypermédia à travers des documents provenant du monde entier, utilisant comme support de communication :INTERNET. Le projet WEB a donc fourni aux utilisateurs branchés sur le réseau INTERNET le moyen d'accéder à des informations variées d'une façon très simple et homogène, quelque soit leur plate-forme.

B. Le fonctionnement du WEB

Le WEB est conçu à partir d'une architecture classique client/serveur. On appelle client, une application qui demande des informations à une autre application qui est susceptible de pouvoir lui fournir. Cette dernière application est appelée : serveur.

Dans le contexte du World Wide Web, un client WEB est appelé browser, il s'agit d'un programme qui permet à un utilisateur de soumettre des requêtes à un serveur WEB et de visualiser le résultat pour le rendre compréhensible par l'utilisateur. Mais un client a aussi la capacité de dialoguer avec d'autres serveurs tels que FTP, TELNET, GOPHER,

... Les services accessibles dépendent du logiciel. Par exemple, Mosaïc (précurseur de tout les browsers, apparu en 1992) peut être considéré comme client « universel ».

Un serveur WEB est un programme qui tourne sur un ordinateur et dont le seul but est de répondre à des requêtes de logiciels clients WEB tournant, en général, sur d'autres ordinateurs. Lorsqu'il reçoit une requête d'un client WEB, il analyse la demande et renvoie l'information demandée ou un message d'erreur au client. Ces requêtes peuvent être simplement la demande de document (ce qui est le plus courant sur le WEB), mais il peut s'agir d'une demande de transfert de fichiers, le résultat de l'exécution d'un programme sur la machine serveur (par exemple le résultat de l'interrogation d'une base de données), ou bien encore, la demande peut même faire l'objet d'un programme à exécuter par le client.

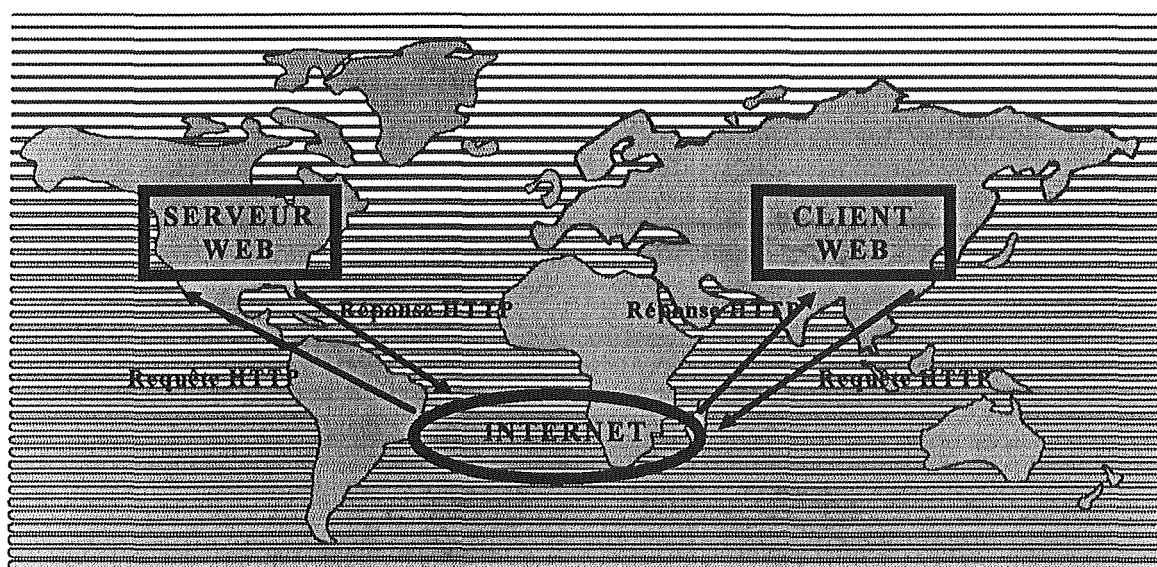


Figure 3.4: représentation d'une requête et sa réponse

Le dialogue entre un client WEB et un serveur se décompose en une série d'étapes bien distinctes, illustrée par la Figure 3.4:

1. Etablissement de la connexion par la requête
2. Le serveur fournit ou non la ressource demandée
3. Fermeture de la connexion

Il n'y a pas de session permanente entre un client WEB et un serveur car la transaction se fait en mode non connecté. Le langage utilisé entre un client WEB et un serveur WEB

s'appelle HyperText Transmission Protocol (HTTP) [Berners-Lee, 1995b]. Ce protocole est décrit en détail dans la suite de ce chapitre: 'HyperText Transfer Protocol'.

Les requêtes de documents hypermédia utilisent le même principe de base. A partir d'un browser (le client WEB), un utilisateur sélectionne un hyper-lien dans un document hypertexte. Le client WEB utilise l'adresse (URL) associée à l'hyper-lien pour se connecter au serveur WEB. Il envoie une requête pour le document concerné au Serveur. Le serveur répond en fournissant au client, le document en question. Il reste au client à afficher le document. Les documents ainsi échangés, sont écrit en HyperText Markup Language (HTML) [Berners-Lee, 1995a]. Ce langage sera décrit dans le point concernant Le langage HTML.

Comme on vient de le mentionner, les documents sont référencés de façon unique grâce aux Uniform Resource Locator (URL) [Berners-Lee, 1994] sur tout le réseau INTERNET. A chaque hyper-lien est associé un URL qui permettra au client d'établir la connexion avec le serveur et de spécifier à celui-ci quel document il désire consulter. La syntaxe d'un URL est standard : méthode ://nom_de_machine :port/nom_de_fichier

La première partie fournit la méthode d'accès au document (file, http, ftp, telnet, wais, news,...) La seconde partie spécifie l'adresse IP de la machine (sous la forme d'une adresse IP ou d'un nom logique DNS). La dernière partie comprend le chemin d'accès à la ressource.

Exemple : `http://www.info.fundp.ac.be/~yph/index.html` (http est la méthode d'accès ; `www.info.fundp.ac.be` est le nom logique du serveur où se trouve le document ; `~yph/index.html` est le chemin d'accès à la ressource concernée).

Les URL peuvent être absolus, c'est-à-dire indiquer exactement l'emplacement d'un fichier sur le serveur mentionné (comme dans notre exemple). Mais il existe aussi la possibilité d'utiliser des URLs relatifs, c'est-à-dire qu'on spécifie soit l'emplacement exact du fichier par rapport au même serveur (sans le mentionné), ou bien , on ne spécifie que le nom d'un document relatif au document dans lequel on se trouve (dans le même répertoire).

IV. HyperText Transfer Protocol

A. ^{0.9}~~HTTP/0.9~~

^{0.9}~~HTTP 0.9~~ est le protocole qui a été défini pour les premières versions du WEB. Il s'agit d'un protocole sans état, assez proche de celui utilisé par Gopher. Il a été conçu pour être léger à utiliser et rapide pour un système hypermédia distribué. Un serveur HTTP écoute les requêtes de clients potentiels sur un port TCP (en général le port 80). L'émission d'un document par un serveur HTTP vers un client WEB demandeur se passe de la façon suivante :

- Le client effectue une connexion TCP sur le port « HTTP » du serveur ciblé, le serveur accepte la connexion ;
- le client émet sa requête, il s'agit d'une simple ligne de texte terminée par les caractères *CR* et *LF*. Le corps d'une requête est composé par la commande *GET* suivie d'un espace et de l'URL du document demandé ;
- le serveur expédie le document demandé, c'est un fichier HTML ;
- le serveur coupe la connexion, elle matérialise aussi la fin du document demandé.

Il est possible d'utiliser des paramètres lors de la requête ce qui permet à la commande *GET* d'interroger un index. Pour cela, on indique les paramètres derrière l'URL, de la façon suivante :

GET URL ?paramètre1+paramètre2+...+paramètreN

B. *HTTP/1.0*

HTTP 1.0 est une évolution de ^{0.9}~~HTTP 0.9~~ et déjà utilisé par presque tous les serveurs WEB. Il consolide le dialogue entre clients et serveurs. Les principaux changements sont les suivants :

- I. extension des façons d'obtenir un document par la création de méthodes d'accès (*GET* n'est plus la seule méthode pour obtenir des documents). Les principales nouvelles méthodes sont les suivantes :
 - A. *Head* : pour obtenir des informations sur un document (date MAJ, nature du serveur HTTP, ...)
 - B. *PUT* : pour mettre à jour la version d'un document
 - C. *POST* : pour créer des nouveaux documents. Elle est utilisée dans l'écriture des procédures de commandes qui nécessitent un grand nombre de paramètres tels que la transmission de formulaire. Ainsi, le contenu d'un formulaire passera par le corps du message et non plus derrière l'URL comme avec la méthode *GET*.
- II. émission de la version de protocole supportée par le client
- III. encapsulation du corps du document à transmettre par des en-têtes MIME.

V. Le langage HTML

HTML est le langage utilisé pour la diffusion de documents hypermédia par les serveurs WEB. Il est basé sur l'utilisation d'un ensemble simple d'étiquettes de formatage de documents. La facilité d'utilisation de ce langage est un des facteurs qui concourt au succès croissant du WEB. HTML et surtout HTML+ permettent la réalisation d'interfaces conviviales.

Les documents HTML sont des documents texte. Il est donc très facile de les créer manuellement à l'aide d'un éditeur de texte ou bien on peut utiliser des utilitaires qui aident à la création. Il existe de nombreux produits sur le marché qui facilitent la création de documents WEB, qu'ils soient textuels ou WYSIWYG. D'autres utilitaires effectuent immédiatement la conversion entre de nombreux formats (Help, RTF, Word, Framemaker, LaTeX,...) et le langage HTML.

Nous allons présenter les bases pour créer un document HTML, mais nous invitons le lecteur à retrouver plus d'informations sur le langage HTML [Berners-Lee, 1995a], [HTML RM, 1996]

A. *Structure générale d'un document HTML*

Voici un exemple de code HTML :

```
<HTML>
<HEAD>
<TITLE>Le titre du document.</TITLE>
<BASE URL="file:///D:/Internet/"></BASE>
</HEAD>
<BODY>
<H1>Level 1 Heading</H1>
<H3>Level 3 Heading</H3>
<P> ceci est le texte d'un paragraphe non formaté </P>
<P> Paragraphe avec mise en forme de caractères: <B>gras</B>,
<I>italique</I>, <FONT SIZE=17>taille</FONT>. </P>
<PRE> Ceci est
    un paragraphe    q u i   e s t    préformaté . . . </Pre>
Pour accéder à la page suivante, <A
HREF="http://localhost/page2.html">cliquer ici</A>.
Insertion d'une image aligné au milieu :<IMG SRC="purpsm.gif"
align=middle>
```



```

<Br>
<UL> liste à puce :
<LI> premier élément de la liste
<LI> le second élément
<LI> et le dernier
</UL>
</BODY>
</HTML>

```

Le document HTML ainsi défini est représenté à la Figure 3.5 tel qu'il serait représenté par un client WEB Netscape.

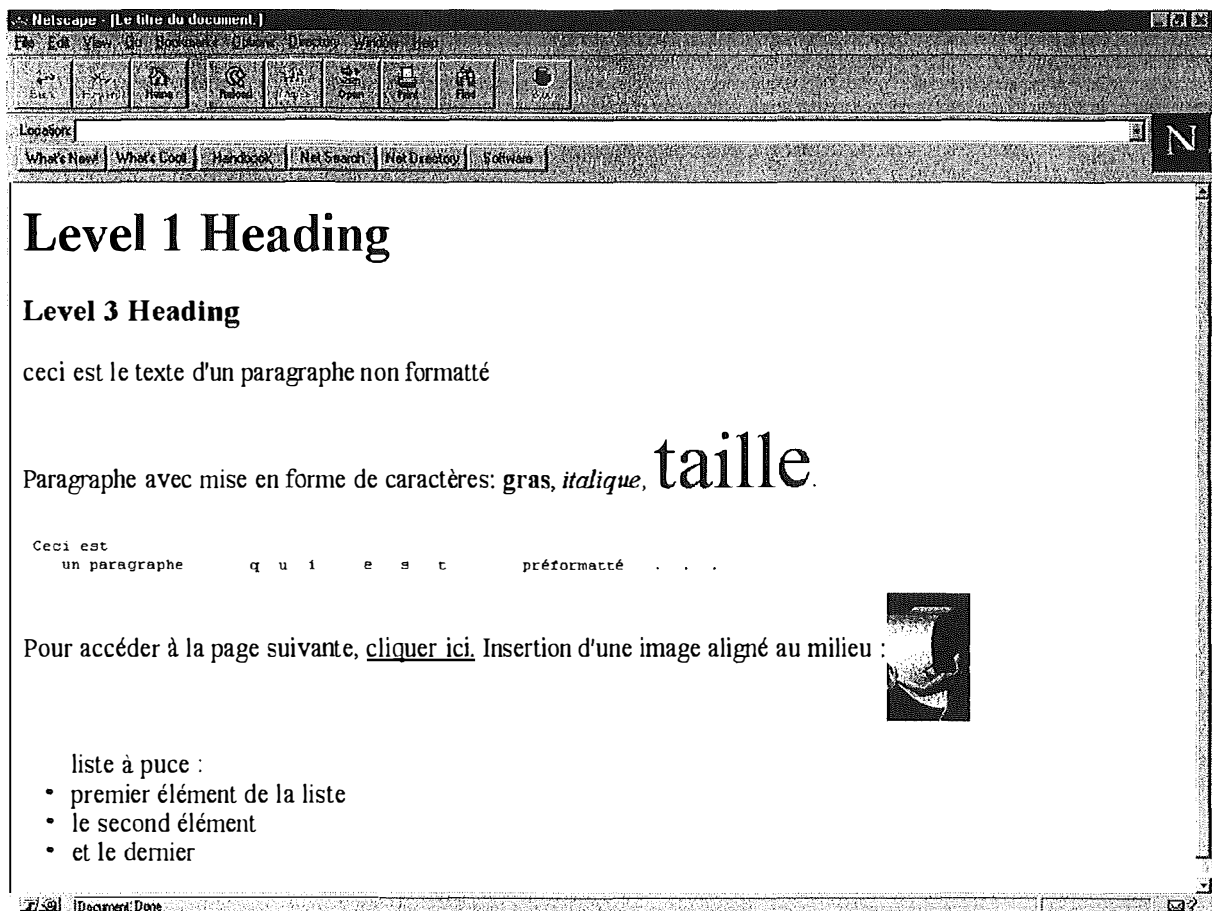


Figure 3.5: exemple de document HTML

Tout document HTML commence par l'étiquette `<html>` et se termine par `</html>`. Ces étiquettes indiquent le début et la fin du document. De manière générale, HTML utilise la même philosophie pour distinguer les différentes parties composant le document. On trouvera donc une étiquette de début et de fin pour délimiter une partie.

Les étiquettes commencent par le symbole `<` et se termine par `>`. Les étiquettes de fin se distinguent des étiquettes de début de partie par le symbole `/` qui suit immédiatement le symbole de début de l'étiquette.

Un document HTML est composé, principalement de deux parties, un en-tête et un corps.

B. Les en-têtes de documents

L'en-tête d'un document est délimitée par les étiquettes `<head>` et `</head>`. Il est composé au minimum d'un titre, mais peut contenir d'autres types d'informations tels que la base du document.

1. Le titre d'un document

Tous les documents HTML devraient contenir un titre. Ce titre apparaît généralement dans une partie séparée de celle du document. Dans notre exemple, le titre apparaît dans la barre titre de la fenêtre. Le titre sert surtout à l'identification du document dans d'autres contextes tels qu'une recherche WAIS.

Le titre d'un document est compris entre les étiquettes `<TITRE>` et `</TITRE>`. Dans notre exemple, il s'agit de :

```
<TITLE>Le titre du document.</TITLE>
```

2. La base d'un document

L'étiquette `<base>` indique l'URL de référence qui sera utilisé pour compléter tous les URLs relatifs dans un document. De telle sorte qu'un document soit encore utilisable même si il est déplacé.

La base d'un document est délimitée par les étiquettes `<base>` et `</base>`. Dans notre exemple, il s'agit :

```
<BASE URL="http://localhost/desserts"></BASE>
```

C. Le corps de documents HTML

Le corps d'un document est délimité par les étiquettes `<body>` et `</body>`. C'est dans cette partie que l'on définit exactement le contenu et la présentation du document HTML, les titres de sections, les paragraphes, les listes, les liens, ...

1. Les titres de sections, les paragraphes et les tabulations

De manière générale, les titres de section sont délimités par les étiquettes `<Hn>` et `</Hn>` où *n* permet de définir le niveau du titre. Il existe six niveaux de titres différents `<H1>` ... `<H6>`. Les différents niveaux de titres sont représentés par des caractères gras de taille décroissante suivant l'importance du niveau.

Exemple :

```
<H1>Level 1 Heading</H1>
<H2>Level 2 Heading</H2>
<H3>Level 3 Heading</H3>
```

Un paragraphe est défini par les étiquettes `<P>` et `</P>`. Il est toutefois possible de ne pas placer l'étiquette de fin. Les espaces, tabulations, retours chariots n'ont pas de signification pour HTML à moins que l'on utilise les étiquettes `<pre>` et `</pre>`.

Exemple :

```
<P> ceci est le texte d'un paragraphe </P>
<PRE> Ceci est
    un paragraphe
    préformaté . . . </Pre>
```

2. La mise en forme des caractères

Pour changer la forme de caractères, deux possibilités s'offrent aux créateurs de documents. Soit ils utilisent des types de caractères prédéfinis que l'on appelle les styles logiques, soit ils utilisent leurs propres styles; on parle alors de styles physiques.

Les styles logiques sont au nombre de sept. On les utilise de la même façon que les autres marquages, c'est-à-dire avec une étiquette au début et une étiquette finale. La représentation des styles logiques est adaptable par les clients WEB. Voici les utilisations proposées; libre à chacun de moduler ces utilisations suivant ses goûts :

- `<DFN>` et `</DFN>` sont utilisés pour matérialiser les définitions
- `<CITE>` et `</CITE>` pour baliser les citations
- `<CODE>` et `</CODE>` pour du code
- `<KBD>` et `</KBD>` pour matérialiser les entrées au clavier

- `<VAR>` et `</VAR>` pour indiquer un nom de variable
- `` , `` , `` et `` pour faire ressortir un texte

Pour les caractères physiques, on a la possibilité de mettre un texte en gras, en italique, à une certaine taille. Pour cela, on utilise respectivement les étiquettes ``, `<I>` et `<SIZE>`. Ces définitions de style ne laissent pas d'alternative de représentation aux clients WEB.

3. Les liens vers d'autres documents

HTML offre la possibilité de créer des liens vers d'autres ressources, d'autres documents. La localisation de ces autres ressources se fait par l'utilisation d'URL. Généralement, les liens textuels sont présentés à l'utilisateur sous la forme de mots souligné et colorés. Ce sont les ancres hypertextes. Il y a différentes possibilités de créer des liens hypertextes.

Les ancres vers des documents externes sont composées de la manière suivante : ` nom de l'ancre `. Dans notre exemple, `cliquer ici`, la représentation de ce lien se fait par le nom de l'ancre ('cliquer ici'). L'URL peut être absolu ou relatif.

On peut définir des ancres à l'intérieur de son document ou vers une partie bien spécifique d'un autre document. Pour qu'une partie spécifique d'un document soit la destination de liens, il faut placer des références dans ce document de la forme ``. Ensuite, il est alors à ce document, ou même aux autres, de définir des liens vers cette référence de la manière suivante : ` nom de l'ancre`. L'URL est optionnel si on la référence est dans le même document.

4. Les images

HTML est plus qu'un hypertexte car il permet aujourd'hui d'inclure différents médias dans un texte. Les images peuvent être soit incluses immédiatement dans le document, soit référencées. Dans le premier cas, on peut décider de l'alignement de l'image : en-dessous, au-dessus ou au milieu de son point d'insertion. La syntaxe est très simple : ``. Par défaut l'alignement est en-dessous. Si l'on désire le changer, on introduit dans l'étiquette, juste après IMG, les mots `ALIGN=TOP`

ou MIDDLE afin de positionner l'alignement respectivement au-dessus et au milieu.

Exemple :

`` définit une image alignée au milieu.

Dans le cas où l'image est référencée, il s'agit uniquement d'indiquer dans l'ancre le chemin d'accès à l'image, son URL. Ce sera uniquement lorsque l'utilisateur décidera de suivre le lien, qu'il verra l'image.

Notons enfin qu'il est possible pour une image de jouer le rôle d'une ancre. Il suffit de structurer son texte de la façon suivante : ` `.

5. Les listes

Le langage HTML permet la création de listes numérotées et de listes avec des puces. Les éléments des listes se placent derrière l'étiquette de début unique ``. Les étiquettes `` et `` entourent respectivement les éléments des listes à puces et numérotées.

Exemple pour la création d'une liste:

```
<UL>
<LI> premier élément de la liste
<LI> le second élément
<LI> et le dernier
</UL>
```

6. Divers

Les étiquettes `
` et `<HR>` permettent respectivement d'insérer un espace horizontal et une barre horizontale afin de structurer d'avantage le document.

VI. Le langage HTML+

HTML+ est une extension de HTML. Il propose des évolutions importantes de HTML pour en améliorer la qualité des documents. Il introduit des notions de figures, de tableaux, de son, d'animation, d'images « réactives », de formules mathématiques, et les formulaires. Nous n'allons présenter dans cette partie que ce qui concerne les formulaires.

A. Les formulaires

Les formulaires sont très utiles car ils permettent le dialogue entre le client et le serveur. En effet, il est possible à un serveur WEB de proposer des documents HTML contenant un ou plusieurs formulaires. Dès lors, il est loisible à l'utilisateur d'un client WEB de remplir le formulaire qui lui est proposé et de le retourner au serveur.

Voici un exemple de document HTML contenant un formulaire et dont la représentation est illustrée à la Figure 3.6.

```
<HTML>
<HEAD>
<TITLE>Customer Newsletter Registration</TITLE>
</HEAD>
<BODY><H1>Customer Registration</H1>
<FORM ACTION="mailto:yph@info.fundp.ac.be">
<P> Please help up to keep you, our customer base, informed about our
products by filling in this questionnaire. When you have entered all
the
information, click on the "Submit Query" button at the bottom. To
register additional users, click on "Reset", then enter them and click
Register again.</P>
<P>Your Name? <INPUT NAME="user" SIZE="30"></P>
<P>Your organization? <INPUT NAME="org" SIZE="30"></P>
<P>How many users at your site? <INPUT NAME="users"></P>
<P>Which Blibbles do you use?</P>
<UL>
<LI><P>MyCompany <INPUT NAME="blibbles" TYPE="checkbox"
VALUE="myco"></P></LI>
<LI><P>XYZ Company <INPUT NAME="blibbles" TYPE="checkbox"
VALUE="XYZ"></P></LI>
```

```

<LI><P>Others <TEXTAREA NAME="competition" COLS="30"
ROWS="4"></TEXTAREA></P></LI>
</UL>
<P>Our Sales and Marketing Team thanks you for your support.</P>
<P><INPUT TYPE="submit"> <INPUT TYPE="reset"></P>
</FORM></BODY>
</HTML>

```

Customer Registration

Please help up to keep you, our customer base, informed about our products by filling in this questionnaire. When you have entered all the information, click on the "Submit Query" button at the bottom. To register additional users, click on "Reset", then enter them and click Register again.

Your Name?

Your organization?

How many users at your site?

Which Blibbles do you use?

- MyCompany ☐
- XYZ Company ☐
- Others

Our Sales and Marketing Team thanks you for your support.

Figure 3.6: exemple de formulaire HTML

1. Déclaration d'un formulaire

Au début d'un formulaire, on précise la nature du traitement qu'il subira, c'est en général l'URL d'une procédure ou d'un programme (se référer à VII Le Common Gateway Interface (CGI)). Si on choisi ce type de traitement, il faut aussi spécifier le type de méthodes avec laquelle le client WEB doit gérer le formulaire pour en renvoyer le résultat au serveur.

Il y a deux types de méthodes pour gérer les formulaires : la méthode POST et la méthode GET. La méthode POST insère les résultats du formulaire dans une variable globale HTTP qui est communiquée au serveur. De son côté, la méthode GET insère

immédiatement les résultats dans l'URL. Il est toujours préférable d'utiliser la méthode POST car elle n'est pas limitative. Par contre, la méthode GET n'autorise l'insertion que de 256 caractères.

La fin d'un formulaire est marquée par une étiquette de fin `</FORM>`. Voici la forme générale de la déclaration d'un formulaire :

```
<FORM METHOD=type de méthode ACTION="URL">
. . . .
</FORM>
```

Il existe une autre manière de traiter les formulaires, sans utiliser de programme CGI. Ce type de traitement demande au client WEB de renvoyer les réponses par E-MAIL à l'adresse définie. Le client WEB compose donc un E-Mail qu'il forme à partir de couples nom de champ - valeur de réponse. C'est cette méthode qui est illustrée dans notre exemple :

```
<FORM ACTION="mailto:yph@info.fundp.ac.be">
. . .
</FOMR>
```

Le corps d'un formulaire est obtenu en intégrant des directives de création de champs à l'intérieur d'un document HTML standard. Chaque champ est défini par l'étiquette `<INPUT>` qui permet de préciser les attributs de ce champ. Il est ainsi possible pour chaque champ de préciser son nom, qui doit être unique pour un formulaire (car sert à identifier le résultat), sa taille, son type, sa valeur par défaut,...

Les points suivant présentent les différents types de champ que l'on peut inclure dans un formulaire.

2. Les champs d'édition

Les champs d'édition unilinéaire et multilinéaire sont acceptés par HTML. Dans tous les paragraphes suivants, un mot en italique est un attribut facultatif. Voici un petit exemple illustrant une utilisation possible des champs d'édition :

```
<P>Nom : <INPUT TYPE=TEXT NAME="nom" VALUE="Dupont" SIZE=20
MAXLENGHT=30></P>
<P>Others <TEXTAREA NAME="competition" COLS="30" ROWS="4">
</TEXTAREA></P>
```


L'argument **VALUE** permet d'insérer un texte par défaut. La taille du champ d'édition est spécifié par l'argument **SIZE**. Le champ **MAXLENGHT** correspond au nombre maximum de caractères que l'on peut entrer dans le champ d'édition. **TEXTAREA** est l'étiquette utilisée pour les champs d'édition multilinéaire. **COLS** définit le nombre de colonnes et **ROWS** le nombre de lignes.

```
<P>Mot de passe : <INPUT TYPE=PASSWORD NAME="mot de passe" SIZE=8  
MAXLENGHT=8> </P>
```

Lorsque l'on insère des caractères dans un champ de type **PASSWORD**, ceux-ci sont cachés par des étoiles ("*").

3. Les boîtes à cocher et les boutons radio

Les boîtes à cocher présentent un certain nombre d'options parmi lesquelles on peut faire un choix booléen. Par contre, dans les boutons radio, on ne peut choisir qu'une seule option parmi toutes celles qui sont proposées.

```
<P>MyCompany <INPUT NAME="blibbles" TYPE="checkbox" VALUE="myco"></P>  
<P>XYZ Company <INPUT NAME="blibbles" TYPE="checkbox" VALUE="XYZ"></P>  
<INPUT TYPE=RADIO NAME="radio 1" VALUE="OK" CHECKED> accord </P>  
<INPUT TYPE=RADIO NAME="radio 2" VALUE="NOK" > désaccord </P>
```

Le nom et la valeur sont utilisés par les CGI pour gérer le résultat. **CHECKED** indique que cette option est sélectionnée par défaut.

4. Les boîtes de sélection

Les boîtes de sélection permettent de choisir un ou plusieurs éléments parmi une liste déterminée à l'avance.

```
<SELECT NAME="PAYS" SIZE=3 MULTIPLE> Pays de destination :  
<OPTION SELECTED> Espagne  
<OPTION> Grèce  
<OPTION> Turquie  
</SELECT>
```

La taille représente le nombre d'éléments qui seront affichés dans la boîte de sélection. L'attribut facultatif **MULTIPLE** signifie que l'utilisateur peut sélectionner plusieurs éléments de la liste. Enfin, si l'attribut **SELECTED** est présent dans l'étiquette **OPTION**, la valeur qui suit sera sélectionnée par défaut.

5. Les boutons de commande

Il existe deux types de boutons de commande; les boutons de soumission et les boutons de remise à l'état initial. Si le bouton de remise à zéro est sélectionné, les champs sont remis à leur valeur par défaut. Le bouton de sélection valide les données insérées et les envoient d'après l'action définie dans l'en-tête du formulaire.

```
<INPUT TYPE=SUBMIT NAME="submit" VALUE="valider">
```

```
<INPUT TYPE=RESET NAME="reset" VALUE="annuler">
```

La valeur correspond à l'intitulé du bouton de commande.

VII. Le Common Gateway Interface (CGI)

Un Common Gateway Interface [CGI 1.1] est un standard pour l'écriture de passerelles entre des serveurs d'informations tel que HTTP et des programmes externes. L'utilisation du CGI permet de rendre les documents HTML plus dynamiques. En effet, un document HTML est toujours composé d'un texte figé. Par contre, les programmes externes sont exécutés en temps réel et ils peuvent de ce fait créer dynamiquement des documents personnalisés à partir des informations reçues. Le rôle de tels programmes externes est de récupérer les paramètres acquis par l'intermédiaire des formulaires, de les traiter et de fournir entre autres un résultat en sortie qui sera envoyé au client WEB qui a soumis la requête.

Par exemple, il est de plus en plus souvent question de s'inscrire avant de pouvoir accéder à un site WEB. Dans ce cas, on reçoit un formulaire à remplir. Suivant les réponses que l'on a fournies, le programme externe acceptera l'inscription, la refusera ou bien lui demandera des informations complémentaires.

Ces programmes externes utilisant le CGI peuvent être écrits dans tout type de langages : C, C++, Fortran, PERL, Unix shell, Visual Basic, AppleScript... Les programmes ainsi écrits sont placés dans le répertoire /cgi-bin de son ^{serveur} ~~browser~~. Le plus souvent on utilisera un langage interprété quand il s'agit d'un script qui doit être souvent modifié.

A. Le lien entre les clients, le CGI et le serveur

Nous allons expliquer dans les paragraphes suivants le fonctionnement des CGI en tant qu'interface entre le client et le serveur.

Le client a devant lui un formulaire HTML qu'il remplit. Les données, ainsi entrées dans le formulaire, sont assemblées par le client WEB en une série de paires nom/valeur. Ces paires sont envoyées au serveur pour être traitées par le CGI spécifié dans l'attribut ACTION de l'étiquette FORM. Comme nous l'avons déjà expliqué, ces couples sont soit directement dans l'URL (dans le cas d'une méthode GET) soit dans une variable HTTP (pour les méthodes POST). Le client WEB envoie alors une requête HTTP contenant ces couples nom-valeur au serveur WEB (étape 1 de la Figure 3.7).

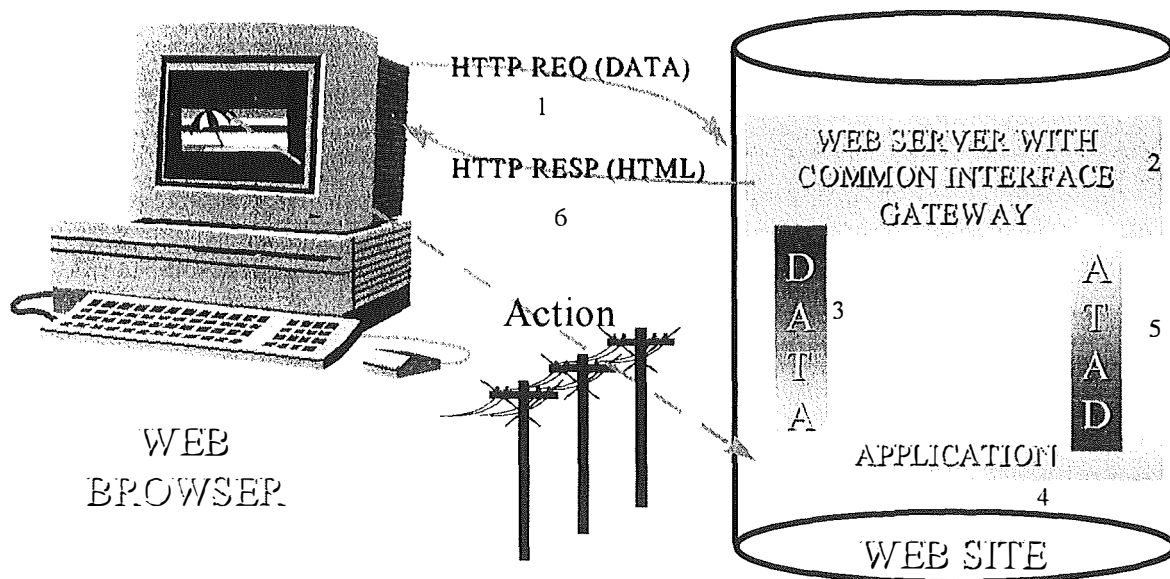


Figure 3.7: interaction client WEB et CGI

Le serveur WEB remarque que l'URL reçu est celui d'un script CGI. Il lance le programme CGI (étape 2 de la Figure 3.7) en lui donnant comme paramètres, sur le canal d'entrée standard, les couples nom/valeur(étape 3 de la Figure 3.7). On peut comparer cette méthode d'exécution au principe de PIPE en UNIX. Le programme CGI consulte à la fois les paramètres, qui lui ont été envoyés sur son entrée standard, et les variables globales de l'environnement dans lequel il s'exécute (Figure 3.8). Cet environnement est parfaitement clos de telle sorte que le programme CGI n'a pas connaissance de l'environnement de travail du serveur. Son environnement est composé uniquement des variables HTTP que le serveur WEB a mis à sa disposition.

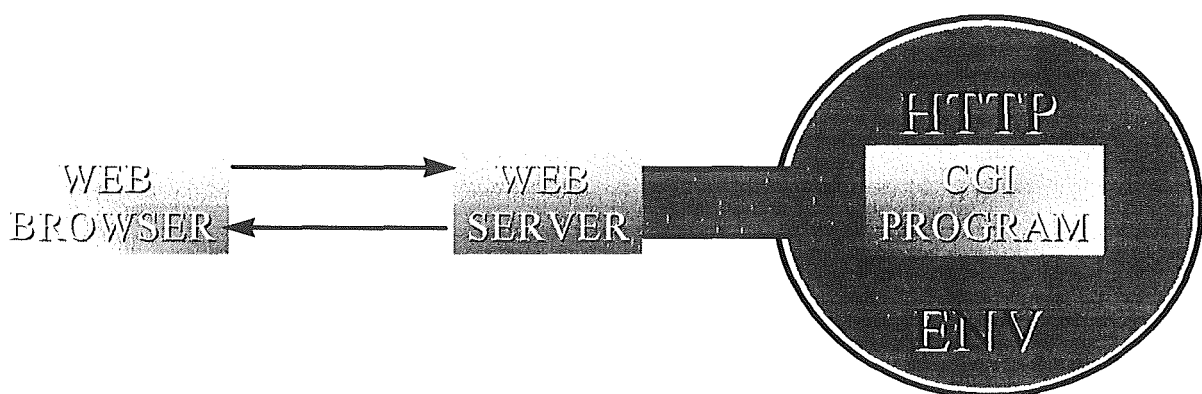


Figure 3.8: environnement d'exécution d'un script CGI

Le programme CGI peut alors effectuer toute une série de tâches telles que la mise à jour ou la consultation d'une base de données, l'envoi d'un mail (étape 4 de la Figure 3.7)... Toutes ces opérations sont exécutées sur le serveur. En fin de compte, le CGI génère un document HTML qu'il envoie sur sa sortie standard (étape 5 de la Figure 3.7). Le

serveur expédie ainsi le résultat du CGI par une réponse HTTP (étape 6 de la Figure 3.7). Le client reçoit et affiche la réponse qui correspond dans la majorité des cas à du code HTML.

Le traitement des paires nom/valeur par le CGI est facilité par des bibliothèques de fonctions qui existent dans tous les langages. Par exemple, le message reçu par le CGI (dans le cas de méthodes GET), est de la forme suivante: `nom_cgi?nom1=valeur1&nom2=valeur2`. Ce message est facilement décomposable.

B. Les variables d'environnement

Les variables d'environnement servent, tout comme les paramètres, à gérer les interactions entre un programme CGI et un serveur WEB. Il existe tout d'abord trois variables qui sont utilisées quel que soit le type de requête :

- `SERVEUR_SOFTWARE` : le nom et la version du serveur qui traite les requêtes.
- `SERVEUR_NAME` : le nom de la machine sur laquelle s'exécute le serveur. C'est l'identifiant utilisé dans la première partie de l'URL soit une adresse IP soit un alias (`info.fundp.ac.be`)
- `GATEWAY_INTERFACE` : la version du CGI sur le serveur.

Suivant les cas, d'autres variables d'environnement interviennent également en voici les principales :

- `SERVER_PROTOCOL` : le nom et la version du protocole utilisé par l'ordinateur qui a effectué la requête (exemple : `HTTP/1.0`).
- `SERVER_PORT` : le numéro de port sur lequel la demande a été reçue.
- `REQUEST_METHOD` : la méthode avec laquelle la requête a été réalisée (exemple : `POST`, `GET`)
- `QUERY_STRING` : les informations qui suivent le point d'interrogation dans l'URL. Cette variable est utilisée par la méthode `GET` qui place les paires nom/valeur dans l'URL à la suite du nom du CGI.

- `REMOTE_HOST` : le nom de l'ordinateur hôte (celui qui fait la requête).
- `REMOTE_ADDR` : l'adresse IP de ce même hôte.
- `CONTENT_TYPE` : les données d'un formulaire de type POST sont placées, toujours sous la forme de paires, dans cette variable.
- `CONTENT_LENGTH` : longueur de la variable `CONTENT_TYPE`.

CHAPITRE 4

LANGAGE DE DESCRIPTION DE SCÉNARIOS POUR SYRECOS

Ce chapitre est consacré aux spécifications des concepts et outils constituant la base du projet ainsi que la définition du rôle de chacun des acteurs intervenant dans l'élaboration d'un scénario.

I. Introduction

Après avoir présenté le contexte général dans lequel le projet s'inscrit, après avoir défini les concepts d'hypertexte et d'hypermédia ainsi que celui de WEB, nous pouvons définir le cadre théorique du projet, ses différentes couches, les différents outils qui composent le projet.

Le but de cet outil est de permettre la génération de scénarios hypermédia à l'aide de modèles de scénarios génériques. Ces scénarios doivent permettre l'utilisation des différentes possibilités offertes par le multimédia ainsi que contenir des formulaires multimédias.

Nous pouvons définir quatre niveaux autour du projet (cfr. Figure 4.9):

■ Le premier niveau

A ce niveau, il est question de développer un atelier de conception. Le but de cet atelier est de mettre à disposition du second niveau, toute une série d'objets et de concepts génériques et utilisables.

Pour cela, l'atelier doit, dans un premier temps, se baser sur un système d'information qui reflète les caractéristiques des scénarios hypermédia en général. Un schéma Entité-Association est défini dans sa globalité avant d'être analysé et expliqué entité par entité.

Dans un second temps, il est intéressant de définir des objets de plus haut niveau, des objets génériques que l'on trouve dans une majorité de scénarios. Ces objets seront définis par rapport au système conceptuel de base.

■ Le second niveau

C'est à ce second niveau que le scénariste définit un scénario générique à l'aide des objets et des outils de l'atelier. Un scénario générique est composé de pages, elles-mêmes constituées d'objets tels que du texte, de la vidéo, du son, des images,... En règle générale, le contenu exact de ces objets n'est pas précisé.

Seul le type de l'objet est déclaré ainsi que certains paramètres propres au type de l'objet tels que sa position, sa description, ...

C'est aussi à ce niveau que la navigation d'enchaînement des pages est créée, ainsi que celle à l'intérieur des pages (par l'entremise des sections). Pour l'aider dans sa tâche, il disposera d'un outil de description de navigation.

■ Le troisième niveau

Ici, le réalisateur choisi le scénario générique qu'il désire instancier. Ce scénario générique lui est fourni par le second niveau. On peut le comparer, par analogie à des modèles de documents. Il navigue de page en page et instancie les objets qui apparaissent sur les pages. Il désigne aussi les paramètres propres à la gestion des formulaires.

■ Le quatrième niveau

Et au dernier niveau, un utilisateur quelconque, visualise les informations du scénario instancié et, le cas échéant, remplit et expédie les formulaires qui lui sont proposés.

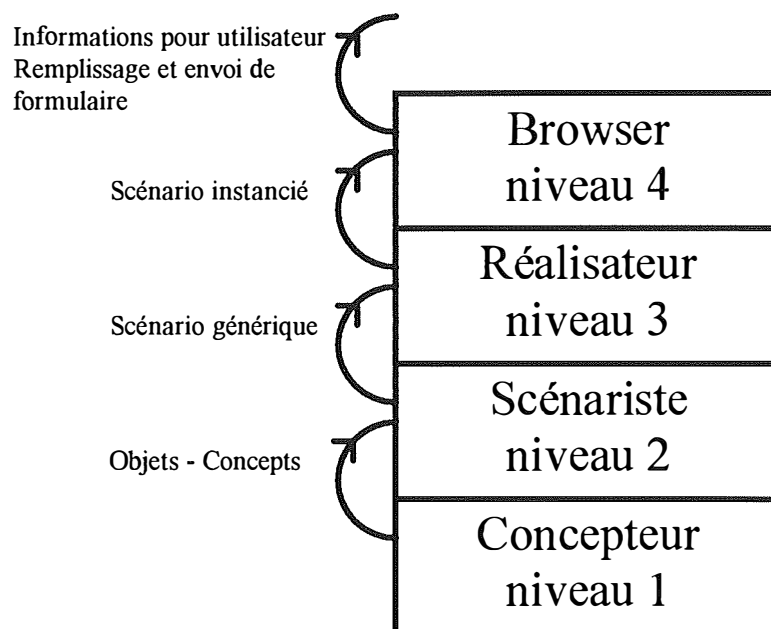


Figure 4.9: découpe en niveaux du projet

Dans la réalité, cette découpe en niveaux permet des retours en arrière entre les différents niveaux. Par exemple, le scénariste peut demander au concepteur de

nouveaux objets, ou bien encore de nouveaux outils pour lui simplifier sa tâche. De même, un réalisateur peut demander au scénariste de modifier certains scénarios génériques.

De même, les différentes couches définies ne sont pas entièrement hermétiques. On peut laisser la possibilité à des réalisateurs plus avancés de définir eux-mêmes une partie de la navigation telle que la navigation intra-pages ou même de définir de nouveaux objets sur une page.

Nous allons maintenant voir plus en détail ces concepts de couches, objets, navigation, ...

II. Schéma conceptuel de base

Nous allons spécifier les concepts de base que le système hypermédia devrait supporter. Ces concepts sont ceux que le scénariste et le réalisateur manipuleront soit directement, ou soit indirectement, par l'intermédiaire d'outils. On peut retrouver une description plus générale de certains de ces concepts dans [Vanderdonckt, 1992a].

Toutes les possibilités offertes par les systèmes hypertextes existant (cfr. Chapitre 2) ne sont pas nécessaire dans le cadre de ce projet. Par exemple, il ne nous intéresse pas de pouvoir suivre un lien d'après une note de musique ou une image précise d'une vidéo.

Les éléments de la Figure 4.10 sont définis ci-dessous en suivant une logique de complexité. Les éléments de base, ou objets de base, sont d'abord définis et viendra ensuite la définition des éléments composés, utilisant ces objets de base.

A. Objets informationnels

Sous le terme d'objets informationnels, il faut entendre tous types d'objets de base contenant de par sa sémantique, une information. Ce sont aussi les objets constituant les caractéristiques du multimédia. Les textes, images, sons, ... sont de tels types d'objets.

Tous les OBJETS INFORMATIONNELS sont identifiés par un *Nom*.

1. Texte

Le TEXTE est un objet informationnel qui sera représenté par une chaîne de caractères dans un certain format (police, couleur, taille, alignement). Cet objet a comme attributs :

- *Alignement* : justification des caractères du texte. Pour garder une certaine cohérence ergonomique, cet attribut est commun pour tout le texte. Il ne sera dès lors pas possible que deux types d'alignement appartiennent au même objet. Il faudra le scinder en deux objets TEXTES distincts. Par défaut, il a la valeur « gauche »



- *Police* : police d'écriture dans laquelle cette PARTIE du TEXTE sera représentée. Par exemple : Times New Roman, Arial, Courier, ... Par défaut, la valeur est « Times New Roman ».
- *Style* : style d'écriture des caractères dans lequel cette PARTIE du TEXTE sera représentée. Par exemple : gras, italique, ... Par défaut, la valeur est « normal ».
- *Couleur* : couleur dans laquelle cette PARTIE du TEXTE sera représentée. Par exemple : bleu, rouge, noir, ... Par défaut, la valeur est « noir »
- *Taille* : taille des caractères composant cette PARTIE du TEXTE. Par exemple : 10, 12, ... Par défaut, la taille est « 12 ».

Un TEXTE est PARTITIONNE en une ou plusieurs PARTIE(s). Chaque PARTIE respectant un numéro d'ORDRE dans ce TEXTE. C'est la composition des CONTENUS de ses PARTIES, qui constitue le contenu informationnel de l'objet TEXTE. Une telle découpe du contenu du TEXTE sera justifié plus loin dans le document, lors de la présentation des liens et du rôle des différents intervenants. Mais d'ores et déjà, cette découpe a l'avantage de permettre aux différentes parties d'un TEXTE d'avoir des formats d'affichage différents. En effet, chaque PARTIE peut se différencier du format général défini par l'objet TEXTE, en y spécifiant une valeur différente pour l'un des attributs *Police*, *Taille*, *Couleur*, *Style*. Toutefois, chacun de ces attributs est facultatif. Seul un attribut différent de ceux de l'objet TEXTE doit être spécifié. Chaque PARTIE d'un TEXTE a les attributs suivants :

- *Contenu* : c'est cet attribut qui contient la chaîne de caractère proprement dit de cette PARTIE du TEXTE.
- *Description* : chaîne de caractères permettant de décrire le contenu sémantique de cette PARTIE du TEXTE.
- *Police* : police d'écriture dans laquelle cette PARTIE du TEXTE sera représentée. Par exemple : Times New Roman, Arial, Courier, ...
- *Style* : style d'écriture des caractères dans lequel cette PARTIE du TEXTE sera représentée. Par exemple : gras, italique, ...

- *Couleur* : couleur dans laquelle cette PARTIE du TEXTE sera représentée. Par exemple : bleu, rouge, noir, ...
- *Taille* : taille des caractères composant cette PARTIE du TEXTE. Par exemple : 10, 12, ...

a) *Exemple* :

si l'on souhaite définir un objet texte comme celui-ci :

« Au sommet, trône le **Moi**, conscient, lucide, raisonnant, personnel et volontaire. Il baigne dans l'**Inconscient personnel** qui contient toutes les expériences *vécues par l'individu*, et qui à son tour, communique avec l'**Inconscient collectif**. Et tout l'édifice repose enfin sur les **instincts profonds**. »⁵

Cet objet devrait être composé de 10 parties. Une partie pour chaque style de caractère.

2. Icône

L'ICONE est un symbole graphique représentant un type d'information, un état quelconque du système, une option, un paramètre, une action, ... Il est caractérisé par les attributs suivants :

- *Contenu* : fichier contenant l'icône.
- *Taille* : taille de l'objet représenté. Cet attribut est composé d'une HAUTEUR et d'une LARGEUR. Par exemple : (640,480), ... Les valeurs sont données en pixels.
- *Description* : chaîne de caractères permettant de décrire textuellement le contenu sémantique de l'objet.

3. Image

L'objet IMAGE permet d'inclure des images, photos... Cet objet est caractérisé par son *Contenu*, sa *Taille* et il possède une *Description*.

- *Contenu* : fichier contenant l'image.

⁵ Pierre DACO, Les triomphes de la psychanalyse, du traitement psychologique à l'équilibre de la personnalité , Bibliothèque Marabout service, 1965

- *Taille* : taille de l'objet représenté. Cet attribut est composé d'une HAUTEUR et d'une LARGEUR. Par exemple : (640,480), ... Les valeurs sont données en pixels.
- *Description* : chaîne de caractères permettant de décrire textuellement le contenu sémantique de l'objet.

La visualisation de ce type d'objet peut être DECLENCHEE de deux manières différentes :

1. *Automatique* : l'image est visualisée de manière automatique, sans action spécifique. Il s'agit du caractère par défaut où aucune spécification supplémentaire de l'objet n'est nécessaire.
2. *Actionné* : l'image est visualisée suite à une action explicite. Pour ce faire, il faut utiliser un artifice visuel pour représenter l'image. Cet artifice servira au DECLENCHEMENT et sera représenté par une ICONE. L'utilisateur devra, s'il désire visualiser l'image, sélectionner cette icône pour l'affichage.

4. Son

Cet objet permet d'inclure des séquences sonores ou de la musique. Il possède comme attribut :

- *Contenu* : le fichier sonore.
- *Description* : elle permet de décrire textuellement le contenu sémantique de l'objet.

L'exécution de ce type d'objet peut être DECLENCHE de deux manières différentes :

1. *Automatique* : la séquence sonore est exécutée sans action spécifique. Il s'agit du caractère par défaut où aucune spécification supplémentaire de l'objet n'est nécessaire.
2. *Actionné* : la séquence sonore est exécutée suite à une action explicite. Pour ce faire, il faut utiliser un artifice visuel pour représenter la séquence sonore. Cet artifice servira au DECLENCHEMENT et sera représenté par une ICONE.

L'utilisateur devra, s'il désire entendre cette séquence, sélectionner cette icône pour déclencher l'exécution de la séquence sonore.

5. Vidéo

Cet objet permet d'inclure des séquences vidéo. La définition de cet objet est très proche de celle du SON. Il possède les attributs suivant :

- *Contenu* : c'est-à-dire le fichier vidéo.
- *Description* : chaîne de caractères permettant de décrire textuellement le contenu sémantique de l'objet.

L'exécution de ce type d'objet peut être DECLENCHE de deux manières différentes :

1. *Automatique* : la séquence vidéo est exécutée sans action spécifique. Il s'agit du caractère par défaut où aucune spécification supplémentaire de l'objet n'est nécessaire.
2. *Actionné* : la séquence vidéo s'exécute suite à une action explicite. Pour ce faire, il faut utiliser un artifice visuel pour représenter la séquence vidéo. Cet artifice servira au DECLENCHEMENT et sera représenté par une ICONE. L'utilisateur devra, s'il désire visualiser cette séquence, sélectionner cette icône pour déclencher l'exécution de la séquence vidéo.

6. Liste

Cet objet permet d'afficher une liste d'articles, les uns en dessous des autres et cela de manière numérotée ou à l'aide de puces. La LISTE peut être de TYPE *numéro* ou *puce*. Une LISTE est composé d'une liste d'ITEMS. Chaque ITEM est un OBJET INFORMATIONNEL. Chaque ITEM possède un numéro d'*ordre* pour la LISTE auquel il appartient. Ce numéro détermine l'ordre des ITEMS dans la LISTE.

a) Exemple :

prenons comme exemple, une partie de définition du sentiment de culpabilité⁶ : «

- sensations d'être toujours fautif

⁶ Pierre DACO, Les triomphes de la psychanalyse, du traitement psychologique à l'équilibre de la personnalité, Bibliothèque Marabout service, 1965

- peur d'être rejeté, blâmé, critiqué
- sensations, fréquentes ou permanentes, d'être rejeté
- soulagement dès qu'on a la sensation d'être pardonné ou accepté
- tout faire pour avoir la sensation d'être pardonné
- le plus souvent, on vit en fonction de l'opinion d'autrui ... »

Cet exemple de liste est composé de 6 ITEMS et est de type *puce*.

7. Tableau

Un TABLEAU permet de représenter des informations selon une certaine organisation. On peut définir un TABLEAU comme un ensemble de lignes et de colonnes où chaque intersection entre une ligne et une colonne s'appelle une cellule. Chaque CELLULE est elle-même un objet INFORMATIONNEL. Toutes les CELLULES possèdent un numéro de colonne (*num_colonne*) et un numéro de ligne (*num_ligne*) qui déterminent la position de la CELLULE dans le TABLEAU. Les attributs du TABLEAU sont :

- *Nbr_Colonne* : définit le nombre de colonnes que possède le TABLEAU
- *Nbr_Ligne* : définit le nombre de lignes que possède le TABLEAU
- *Description* : chaîne de caractères permettant de décrire textuellement le contenu sémantique de l'objet.

a) Exemple :

voici l'exemple d'un tableau où chaque cellule définie est un objet de type TEXTE :

	Hauteur moyenne
Homme	1,9
Femme	1,7

Ce tableau est composé de 3 lignes (*Nbr_Ligne* = 3) et de 2 colonnes (*Nbr_Colonne* = 2). La cellule de la première ligne (*Num_Ligne* = 1) et de la première colonne

(Num_Colonne = 1) n'est pas défini. La cellule de la seconde ligne et seconde colonne contient un objet TEXTE contenant « 1,9 ».

B. Objet interactif

Sous le terme d'objet interactif, il faut entendre tout objet de dialogue utilisé pour la saisie et / ou l'affichage d'informations relatives à la constitution d'un formulaire. Les EDITBOX, LISTBOX, TOGGLEBUTTON,... sont de tels types d'objets. Il s'agit de la seconde et dernière catégorie des objets de base.

Tous les objets interactifs sont identifiés par leur *Nom* et peuvent posséder un *Libellé*. Le *Libellé* est un intitulé textuel simple que l'on place pour définir le contenu informationnel d'un objet ou pour préciser la signification de cet objet.

Tous les objets interactifs possèdent un numéro (*tab_orders*) par formulaire qui détermine l'ordre des objets dans ce formulaire. Cela permettra de définir une navigation entre les objets interactifs du formulaire.

Tous les objets interactifs contiennent un attribut *Description* permettant de définir sémantiquement l'objet.

1. EditBox

Il s'agit d'un champ d'édition unilinéaire. Il permet à l'utilisateur d'introduire et de manipuler des chaînes de caractères en utilisant le clavier. Il se représente par un libellé (si défini) et une zone rectangulaire dans laquelle le texte peut être introduit, ainsi que d'une invitation à saisir.

Cet objet a comme attribut :

- *Long_Max* : spécifie la longueur de la chaîne maximale de caractères que peut contenir le champ d'édition unilinéaire.
- *Long_Champ* : spécifie la longueur de la chaîne maximale de caractères que le champ peut afficher en une fois tel que $1 \leq \text{long_champ} \leq \text{long_max}$.

- *Type_Donnée* : spécifie le type de syntaxe des caractères qui peuvent être saisis et / ou affiché dans le champ d'édition unilinéaire. Par exemple : le type *texte*, *password*,...
- *Valeur* : spécifie la valeur initiale qui sera affichée dans le champ d'édition unilinéaire. Cet attribut n'est pas obligatoire.

a) *Exemple :*

Voici la représentation d'une EDITBOX de type TEXTE ayant la valeur textuelle « *Valeur de l'EDITBOX* » suivi de l'invitation à saisir :

Valeur de l'EditBox |

La *Long_Max* est de 30 et la *long_champ* est de 23.

2. MultiEditBox

Il s'agit d'un champ d'édition multilinéaire. Il permet à l'utilisateur d'introduire et de manipuler des chaînes de caractères en utilisant le clavier. Tout comme l'EDITBOX, il se constitue d'un libellé (si défini), d'une invitation à saisir et d'une zone rectangulaire dans laquelle le texte peut être introduit. Contrairement au champ d'édition unilinéaire, la zone rectangulaire peut s'étendre sur plusieurs lignes et permet non seulement un défilement horizontal mais aussi vertical.

Cet objet a comme attribut :

- *Long_Max* : spécifie la longueur de la chaîne maximale de caractères que peut contenir le champ d'édition multilinéaire.
- *Nbr_Ligne* : spécifie le nombre de lignes que le champ d'édition multilinéaire peut afficher
- *Nbr_Colonne* : spécifie le nombre de colonnes que le champ d'édition multilinéaire peut afficher
- *Valeur* : spécifie la valeur initiale qui sera affichée dans le champ d'édition multilinéaire. Cet attribut n'est pas obligatoire

a) *Exemple :*

Voici la représentation d'un MULTIEDITBOX ayant la valeur textuelle « Texte contenu dans le MULTIEDITBOX ». La long_max est de 256, le nombre de lignes est de 6 et le nombre de colonnes est de 25.

Texte contenu dans le
MultiEditBox

3. ListBox

La LISTBOX est une liste de sélection déroulable d'éléments représentant des choix parmi lesquels l'utilisateur a la possibilité de faire une ou plusieurs sélections. Les éléments composant la liste de choix s'appellent des ITEMS. La LISTBOX est représentée par une zone rectangulaire contenant les ITEMS de la liste présentée les uns en dessous des autres, et une barre de défilement à droite de la zone rectangulaire. Sa longueur dépend de la longueur des ITEMS.

Cet objet a comme attribut :

- *Multiple* : spécifie s'il est possible de sélectionner plusieurs Items de la liste de choix.
- *Nbr_Visible* : spécifie le nombre d'Items visibles en même temps dans la liste de sélection.

a) *Exemple :*

Voici l'exemple d'une LISTBOX composé d'une liste de 6 items dont le nbre_visible est 4.

item 1
item 2
item 3
item 4

4. ComboBox

La COMBOBOX est une variante de la LISTBOX où la liste de sélection reste toujours cachée, à moins que l'utilisateur ne le demande. Ce genre d'objet interactif est utilisé pour éviter la saturation de la présentation à l'écran. Il ne permet pas la sélection multiple et ne propose donc qu'un seul item visible à la fois. Toutefois, la COMBOBOX dispose d'une fiche de défilement qui déclenche l'apparition de la liste de sélection. Tout comme la LISTBOX, ces éléments de sélection sont des ITEMS.

a) *Exemple :*



5. ToggleButton

Le TOGGLEBUTTON est un mécanisme de sélection représenté par un groupe de boutons de sélection. Il propose des éléments de choix (ITEMS) sous la forme d'une liste d'items ou chaque item est précédé d'un bouton qui peut être sélectionné ou désélectionné par l'utilisateur. Le TOGGLEBUTTON est représenté par une liste composée des *Libellés* d'ITEM. A chaque libellé correspond un indicateur de sélection.

Ces attributs sont :

- *Type* : spécifie le type de TOGGLEBUTTON. Le type peut être soit « *check* », soit « *radio* ». Dans le cas où le type est « *check* », les choix d'items ne sont pas mutuellement exclusifs et à tout moment, aucune, une ou plus d'une boîte à cocher peuvent être sélectionnées. L'indicateur de sélection est représenté par un carré. Dans le cas où le type est « *radio* », l'utilisateur peut faire un et un seul choix parmi les ITEMS proposés. Les choix sont mutuellement exclusifs et, à tout moment, un et un seul choix est sélectionné. L'indicateur de sélection est représenté par un cercle.
- *Alignement* : spécifie si les ITEMS sont alignés de manière horizontale ou verticale.

a) *Exemple :*

Ce premier exemple illustre un TOGGLEBUTTON composé de 3 items et est de type *check*. Le mode d'alignement est *vertical*. Deux items ont été sélectionnés.

☒ CheckBox1
☐ CheckBox2
☒ CheckBox3

Le second exemple illustre un TOGGLEBUTTON composé de 3 items et de type *radio*. Le mode d'alignement est *horizontal*. Un seul item peut être sélectionné.

☐ RadioButton1 ☒ RadioButton2 ☐ RadioButton3

C. Section

Une SECTION est composée d'OBJETS INFORMATIONNELS et possède un *Nom* identifiant. Les objets qui composent la SECTION ont un numéro d'*Ordre* et un *Alignement* dans cette SECTION. Le numéro d'*Ordre* détermine la hauteur relative de la position de l'objet dans la section. L'objet d'ordre 3 sera positionné en-dessous de l'objet d'ordre 2. Une fois la hauteur de l'objet déterminé, on peut l'aligner à *gauche*, au *centre* ou à *droite*. Ceci permet de définir un alignement logique de l'objet.

Ce mécanisme de positionnement relatif est nécessaire étant donné que tous les objets n'ont pas une taille fixe connue dès le départ. Si deux objets ont le même numéro d'ordre, ils seront positionnés, dans la mesure du possible, au même niveau. Cela dépendra, en outre, de la taille des objets.

La SECTION contient un attribut DESCRIPTION qui permet de contenir une définition sémantique de cette SECTION.

La SECTION n'a aucune représentation graphique, si ce n'est la représentation des objets qui la compose.

D. Formulaire

Le FORMULAIRE pourrait se définir comme une SECTION interactive. Le FORMULAIRE hérite des propriétés de la SECTION. Cela signifie que tout comme la SECTION, le FORMULAIRE est identifié par un *Nom* et possède une *Description* et il est aussi composé d'OBJETS INFORMATIONNELS. En plus, il est composé d'OBJETS d'INTERACTIFS. Ces objets ont aussi un numéro d'*Ordre* et un *Alignement* dans le FORMULAIRE.

Enfin, le FORMULAIRE doit être envoyé par un objet SUBMIT et peut aussi ETRE ANNULE PAR un objet RESET.

E. Submit

L'objet SUBMIT permet de définir les paramètres logiques de l'envoi d'un FORMULAIRE. Il contient un *Libellé*, un *Script* et une *Méthode*. La *Méthode*⁷ permet de définir la méthode de postage des informations contenues par les objets interactifs. Le

⁷ Par exemple, dans les spécifications HTML, la *Méthode* peut être soit « *get* » ou « *post* ».

Script quant à lui, permet de définir l'action à exécuter lors de l'envoi du formulaire rempli. Il peut s'agir de localiser le script CGI qui sera exécuté lors de la réception des données postées.

Il est représenté par un bouton de commande dont le libellé correspond à l'attribut *Libellé*, s'il existe, ou sinon, il aura par défaut la valeur « *submit* ».

F. Reset

L'objet RESET a pour rôle d'effacer toutes les données introduites par l'utilisateur. Il remet le formulaire dans son état initial. Il est représenté par un bouton de commande dont son libellé peut être spécifié à l'aide de l'attribut *Libellé* de RESET. Par défaut, cette valeur est « *reset* ».

G. Page

Une PAGE est découpée en une ou plusieurs SECTIONS et/ou propose un ou plusieurs FORMULAIRES. Ces SECTIONS et FORMULAIRES ont un numéro d'*Ordre* dans la PAGE. Une PAGE ne peut contenir deux éléments avec le même numéro d'*Ordre* mais doit contenir au moins une SECTION ou un FORMULAIRE. La PAGE est identifiée par son *Nom* et possède un *Titre*.

Cet objet contient un attribut DESCRIPTION qui permet de contenir une définition sémantique de l'objet.

a) Exemple :

La Figure 4.11 illustre un objet PAGE découpé en deux sections. La première section comprend un objet TEXTE et un objet IMAGE. Quant à la seconde section, elle comprend un objet TABLEAU de 2 lignes et 6 colonnes.

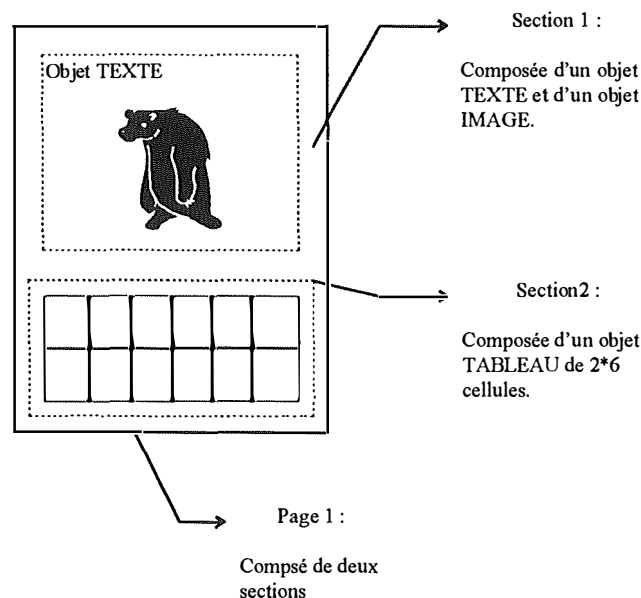


Figure 4.11 : Représentation d'une PAGE

H. FrameSet

Le FrameSet permet de diviser un écran en plusieurs régions, appelées FRAME, ce qui permet de représenter plusieurs pages, en même temps, à l'écran. Implicitement, une page est synonyme d'écran. Avec les FRAMES composant un FRAMESET, on peut représenter plusieurs pages, en même temps, à l'écran. Ce qui permet d'afficher des informations d'une manière plus utile et plus flexible.

Il est donc possible d'avoir des éléments toujours visibles tels que des barres de contrôle, des titres graphique. Par exemple, les FRAMES peuvent contenir des objets de navigation indépendants, tels que des tables des matières.

Le FRAMESET est identifié par un *nom*. On définit la proportion des *Colonnes* à l'écran, pour cela on déclare une liste de pourcentage. De cette manière, le nombre de *colonnes* est connu implicitement grâce au nombre d'éléments dans la liste. Pour chaque Colonne, on spécifie une liste du même type pour définir les proportions des *Rangées*.

Chaque partie d'écran définit un FRAME. Le FRAME peut servir à ne représenter qu'une seule page statique ou bien un ensemble de pages reliées entre elles.

a) *Exemple :*

Voici un exemple de FrameSet (Figure 4.12) décomposé en 5 Frames. Les deux *colonne* ont été définies par la liste « 50%,50% ». Et, de manière respective pour les rangées : « 50%,50% » et « 33%, 33%, 33% ».

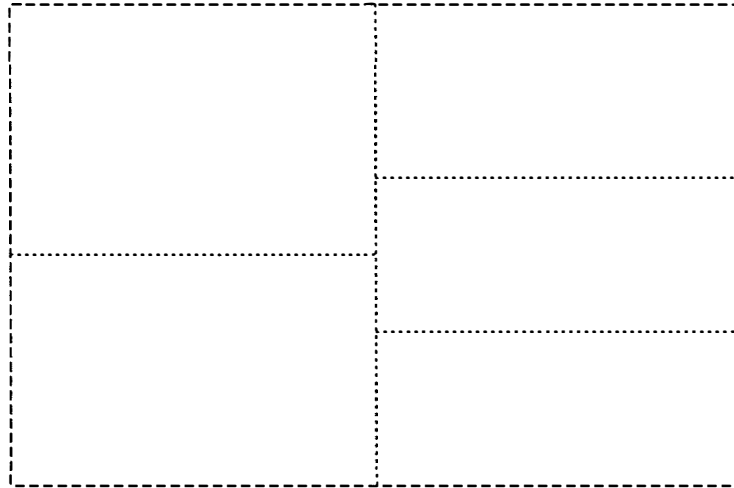


Figure 4.12: Exemple de frameset

I. Lien

Un LIEN possède une et une seule SOURCE et une et une seule DESTINATION. La SOURCE peut être soit une PAGE, une SECTION, un FORMULAIRE, une PARTIE d'un TEXTE, une ICONE, ou encore, une IMAGE. La DESTINATION peut être soit une autre PAGE, une autre SECTION ou un autre FORMULAIRE que la SOURCE.

Un lien possède un *Libellé*, un *Type* et une *Position*. Le *Libellé* correspond au texte qui représentera le lien sur la page. La *Position* permet de déterminer l'emplacement des LIENS sur une PAGE. Ils peuvent être en début de PAGE, en fin de PAGE ou encore en début et en fin de PAGE⁸. Dès lors, la valeur de *Position* peut être « *début* », « *fin* » ou bien « *début-fin* ». Quant au *Type*, il sert à définir le type de représentation du LIEN si celui-ci a comme source une PAGE ou une SECTION. Ce *type* peut avoir la valeur « *texte* » pour une représentation textuelle du LIEN ou bien, il peut avoir la valeur « *bouton* » pour une représentation sous forme d'un bouton.

Cet objet contient un attribut *Description* qui permet de contenir une définition sémantique de l'objet.

⁸ Même caractéristique pour les LIENS appartenant à une SECTION ou un FORMULAIRE

J. Scénario

Un SCENARIO est identifié par un *Nom* et est constitué d'un ensemble de PAGES. Il contient aussi un attribut *Description* permettant de donner une définition sémantique du SCENARIO ainsi qu'un TITRE.

Le SCENARIO possède obligatoirement une et une seule PAGE DEPART. C'est la première PAGE du SCENARIO.

a) Exemple :

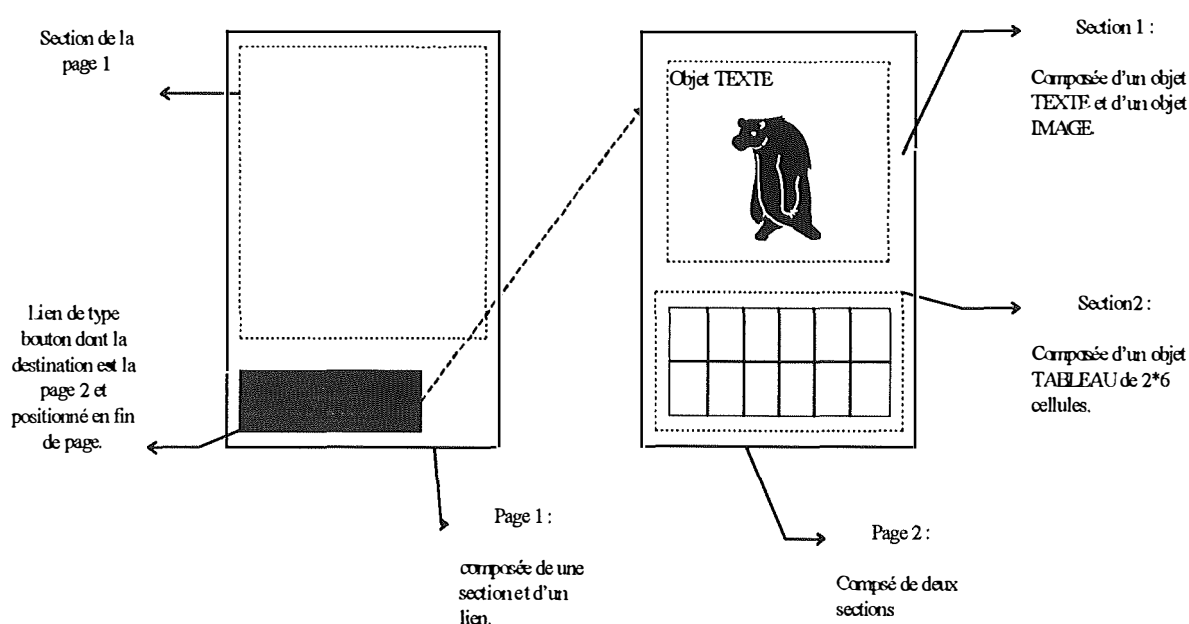


Figure 4.13 : Exemple de scénario générique

La Figure 4.13 illustre un exemple de scénario simpliste, composé de deux pages reliées par un lien. Ce Lien est de type « *bouton* », il est positionné en fin de page et à comme libellé « *aller à la page 2* ».

K. Conclusion

Tous les objets de base viennent d'être analysés. Il serait intéressant de définir comment ces objets peuvent être utilisés dans les couches supérieures. Quels sont les rôles assignés à chacun des intervenants?

II. Rôle du scénariste

Le scénariste est la personne qui créera un scénario générique à partir des objets du système de base qui sont à sa disposition. Un scénario générique est un scénario hypermédia, c'est-à-dire un ensemble de pages reliées entre elles, où les objets de base composant ces pages sont génériques. Cela signifie que lors de la création d'un scénario générique, le scénariste doit surtout définir quels sont les objets composant les pages ainsi que leur positionnement. Il s'occupe principalement de la composition et de l'agencement général des objets sur les pages, et non du contenu de ces objets. En général, seule la position et le type d'objet sont spécifiés. Le contenu final des objets sera instancié plus tard par le réalisateur. De façon imagée, le scénariste établit les fondements, le squelette du scénario.

Toutefois, il aura aussi la possibilité, s'il le désire, de définir complètement les objets et leur contenu et cela dans un souci de conservation de la cohérence ergonomique des scénarios. Par exemple s'il désire que chacune des pages contienne la même note en bas de page, ou bien le même logo.

Le scénariste doit donc disposer de fonctionnalités pour développer son scénario générique. Il lui faut une manière d'utiliser les objets de base du système. Il pourrait disposer d'un langage de description pour définir son scénario. La définition d'un scénario pourrait être textuelle ou même visuelle (WYSIWYG). Mais quel que soit le type de définition choisi, il devrait respecter le même langage de base, telle que le langage basé sur la grammaire BNF, définie dans l'ANNEXE 1.

Cette grammaire précise formellement la forme que pourrait prendre les spécifications mais elle ne sera jamais visible par les différents intervenants qui travailleront principalement en WYSIWYG.

Pour définir de nouveaux scénarios génériques, le scénariste l'identifiera en lui donnant un nom et une description. Il devra ensuite définir l'ensemble des pages appartenant au scénario ainsi que la navigation entre les pages. La première page définie sera la page de départ du scénario.

Le scénariste a aussi la responsabilité de définir entièrement les Framesets. Il lui donne un nom, définit la découpe de l'écran en Frames. Il doit aussi spécifier le contenu de chaque Frame.

Pour chaque page, le scénariste doit en donner une description, un titre et a la possibilité de la composer entièrement. Il définit le ou les sections composant la page. L'ordre dans lequel les sections d'une page seront définies, déterminera l'ordre que suivra les sections dans cette page. Il donnera une description à la section. Tous les objets composant une section doivent ensuite être déclarés ainsi que leur numéro d'ordre et leur alignement logique et cela pour toutes les sections. Si le scénariste souhaite utiliser un formulaire sur sa page, il le déclare de la même manière qu'il déclarerait une section. Ensuite, il devra obligatoirement définir un objet SUBMIT et s'il le souhaite, un objet RESET avant la liste de tous les objets composant le formulaire. Chaque objet a une déclaration particulière et un certain niveau de liberté peut être laissé au réalisateur.

- TEXTE

Le scénariste donne un nom à l'objet TEXTE. Il peut déterminer l'alignement ainsi que le format général du texte de l'objet ou le laisser au choix du réalisateur. Il détermine aussi la ou les PARTIES de TEXTE. Pour chaque PARTIE, il peut en définir le contenu ou bien donner une description de ce que devra contenir la PARTIE. Il a aussi la possibilité de définir un ou plusieurs des paramètres du format du TEXTE. Il peut aussi déterminer si une partie du TEXTE est source d'un lien ou pas. Pour cela, il lui suffit de donner le nom identifiant de la page ou section cible. S'il désire définir un lien qui sera instancié par le réalisateur, il ne spécifie pas d'*identifiant* à la destination.

- LISTE

Le scénariste donne un nom identifiant à la liste ainsi qu'une description. Il peut déterminer le type de liste (puce ou numéro) et les items composant la liste, ou bien laisser ce choix au réalisateur. Pour définir les items, il décrit un objet informationnel par items. L'ordre des items est défini par l'ordre dans lequel des objets informationnels sont définis.

- TABLEAU

Le scénariste détermine un nom identifiant pour le TABLEAU et en donne une

description. Son nombre de colonnes et/ou son nombre de lignes peuvent être défini par le scénariste. S'ils ne sont pas définis, ils sont laissés au choix du réalisateur. Si le scénariste les a définis, il peut définir certaines cellules du tableau. Pour cela, il désigne l'objet informationnel de la cellule qu'il aura sélectionné par le numéro de ligne et le numéro de la colonne.

- ICONE

Le scénariste détermine un nom identifiant pour l'icône et en donne une description. Sa taille peut être spécifiée par le scénariste. Il peut aussi déterminer si l'icône est source d'un lien ou nom en spécifiant le nom de la page ou de la section cible. S'il désire définir un lien qui sera instancié par le réalisateur, il ne spécifie pas d'*identifiant* à la destination.

- IMAGE

Le scénariste détermine un nom identifiant pour l'image et en donne une description. Il peut spécifier la taille de l'image, déterminer si l'image est source d'un lien ou nom en utilisant le nom de la page ou de la section cible. Le scénariste peut spécifier l'icône qui sera la source de déclenchement de l'image. Si aucune icône n'est spécifiée, l'image aura un déclenchement automatique. Si une icône joue un rôle déclencheur, elle ne peut être source d'un lien. S'il désire définir un lien qui sera instancié par le réalisateur, il ne spécifie pas d'*identifiant* à la destination.

- SON

Après lui avoir donné un nom et une description, le scénariste peut spécifier l'icône qui sera la source de déclenchement du son. Si aucune icône n'est spécifiée, le son aura un déclenchement automatique. Si une icône joue un rôle déclencheur, elle ne peut être source d'un lien.

- VIDEO

Après lui avoir donné un nom et une description, le scénariste peut spécifier l'icône qui sera la source de déclenchement de la vidéo. Si aucune icône n'est spécifiée, la vidéo aura un déclenchement automatique. Si une icône joue un rôle déclencheur, elle ne peut être source d'un lien.

- EDITBOX

Comme pour tous les objets interactifs, le scénariste donne un nom identifiant à l'EDITBOX, une description, un numéro de tab_order sur le formulaire, et peut spécifier un libellé. De plus, le scénariste doit spécifier tous les autres attributs. Il doit définir le nombre de caractères que peut contenir l'objet, le nombre de caractères visibles (largeur du champ), le type de données ainsi que la valeur initiale désirée.

- MULTIBOX

Le scénariste doit donner un nom identifiant à l'objet ainsi qu'une description, un numéro de tab_order sur le formulaire, et définit la longueur maximale du champ, le nombre de lignes et le nombre de colonnes. Il peut aussi initialiser le champ et lui donner un libellé.

- LISTBOX

Le scénariste spécifie le nom de l'objet, sa description, un numéro de tab_order sur le formulaire, ainsi que son libellé éventuel. Il doit aussi en définir la représentation, à savoir le nombre d'items visibles. Il faut définir le type de sélection (MULTIPLE) et la liste d'items. S'il définit les items de la LISTBOX, l'ordre dans lequel ils sont définis, définit l'ordre d'apparition des items dans l'objet. Pour chaque item qu'il définit, le scénariste doit spécifier s'il est sélectionné ou pas. S'il sélectionne plusieurs items alors que le type de la LISTBOX ne permet qu'une sélection, seul le premier item désigné comme sélectionné le sera effectivement.

- COMBOBOX

Pour cet objet, seul le nom, la description et un numéro de tab_order sur le formulaire doivent obligatoirement être définis. Le scénariste peut toutefois en spécifier le libellé ainsi que la liste d'items.

- TOGGLEBUTTON

Le scénariste donne un nom au groupe de boutons, la description de l'objet, un numéro de tab_order sur le formulaire et il peut lui associer un libellé. Il spécifie le type de bouton qu'il désire selon les caractéristiques qu'il en attend. Il choisira le type « radio » s'il ne désire qu'une et une seule réponse, « check »

dans les autres cas. En conclusion, le scénariste a pour objectif de s'occuper de la présentation générale du scénario, il doit aussi spécifier le type d'alignement des boutons. Finalement, il peut s'il le désire définir la liste des boutons.

Une fois les pages et leurs objets composants définis, le scénariste doit définir la navigation entre ces pages. Pour cela, il peut spécifier des liens. Le scénariste leur donne un nom, une source et une destination. Il doit aussi déterminer la position sur cette source, à savoir « début » ou « fin » ou « début fin » ainsi que le type. Il peut aussi en fixer le libellé ou en donner une description.

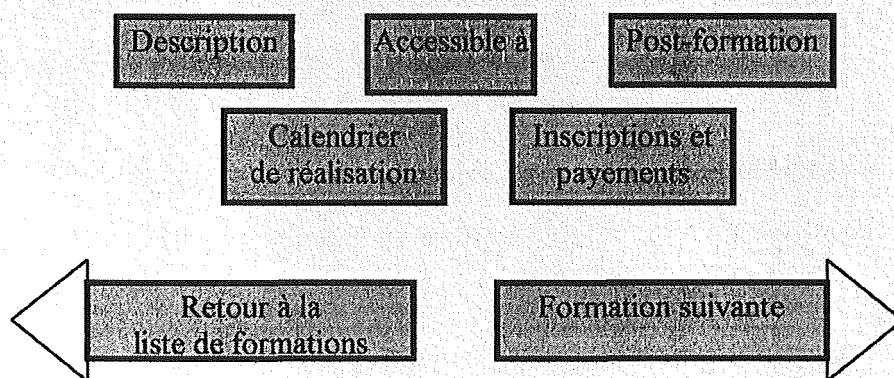
III. Exemple de scénario générique

Voici un exemple de scénario générique pour le projet SYRECOS. Ce scénario est valable pour la présentation d'une formation. Cet exemple se base uniquement sur une présentation d'informations textuelles. Toutefois, il serait tout à fait possible d'y inclure d'autres objets de présentation comme des images, du son, ...

Ce scénario est composé de 6 pages. Chacune de ces pages permet de se rendre à toutes les autres pages. Les liens « retour à la liste de formation » et « formation suivante » sont des liens provenant d'une structure englobant cet extrait de scénario.

Identification

- **Titre** : <titre de la formation>
- **Réf** : <identifiant de la formation>
- **Type de formation** : <type de formation⁹, mode de formation¹⁰>
- **Durée** : <nombre de périodes>, <type de période¹¹>
- **Langue** : <liste des langues dans laquelle peut être donnée la formation>
- **Prix** : <montant avec TVA> TVAC (soit <montant sans TVA> HTVA)¹²
 <commentaire-prix>
 <Réductions possibles>
- **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>
- **(Cette formation fait partie du programme global : <Titre du Programme>)¹³**



⁹ A priori les types de formation retenus sont : Cours, Conférence.

¹⁰ De manière générale, les modes de formation retenus a priori sont : "Intra", "Inter", "Inter avec possibilité intra", "Correspondance", "Téléphone". Le scénario de présentation proposé ici ne convient toutefois que pour les modes "Inter" et "Inter avec possibilité intra". D'autres scénarios plus pertinents seront proposés pour les autres modes.

¹¹ A priori, les types de périodes seront "soirée(s)", "1/2 jour(s)", "jour(s)", "séances" (pour les cours téléphoniques ou par correspondance).

¹² Typiquement belge

¹³ Si la formation fait partie d'un "programme global" (groupe de formations distinctes proposées comme un "produit" unique).

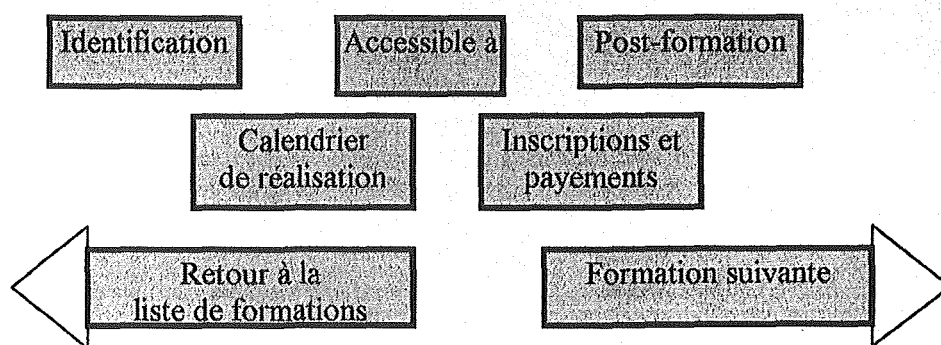
Description

- **Titre** : <titre de la formation>
 - **Durée** : <nombre de périodes, type de période>
 - **Type de formation** : <type de formation, mode de formation>
 - **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>
- =====
- **Objectifs** : <texte libre de spécification des objectifs>
 - **Description** : <texte libre d'introduction à la formation>
 - **(Animateur(s))¹⁴** : suite de {<nom, prénom animateur>, (<CV animateur>)})
 - **Contenu** : suite de {<titre de partie >,

(texte libre décrivant le contenu abordé dans cette partie)¹⁵,

(liste de {<nom, prénom animateur>, (<CV animateur>)})¹⁶

}
 - **Méthodes pédagogiques** : <texte décrivant les méthodes utilisées>
 - **Nombre maximum de participants** : <nombre> ou "non-limité"
 - **Pour plus d'information, contacter** : <Nom de contact>, soit directement à l'adresse <adresse-mail> ou au n° de téléphone <n° de téléphone>.



¹⁴ Si les animateurs sont spécifiés pour chaque partie de la formation (ex : conférence), la liste des formateurs n'apparaîtra pas ici.

¹⁵ Ce texte peut être structuré en niveaux (2 maximum). Voir annexe 1 : Présentation d'un contenu structuré.

¹⁶ Dans le cas où il est préférable d'associer à chaque partie, son ou ses animateur(s) spécifique(s). Dans ce cas, les animateurs ne seront pas associés à la formation (voir note de bas de page précédente)

Accessible à

- **Titre** : <titre de la formation>
- **Type de formation** : <type de formation, mode de formation>
- **Durée** : <nombre de périodes, type de période>
- **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>

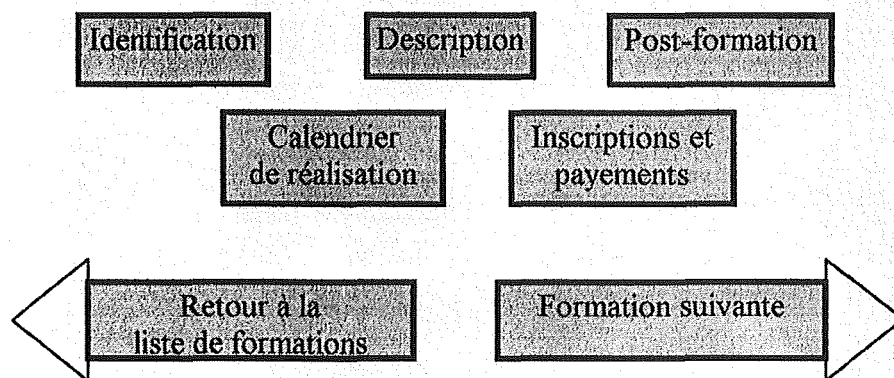
=====

- **Public visé** : <liste des public-cibles>¹⁷

- **Connaissances prérequis** : <texte libre>

ou¹⁸ "Un niveau de connaissance comparable à celui pouvant être acquis par les formations : <listes des formations prérequis> est **indispensable**".

ou¹⁹ "Un niveau de connaissance comparable à celui pouvant être acquis par les formations : <listes des formations prérequis> est recommandé".



¹⁷ La typologie des publics cible est présentée en annexe 2.

¹⁸ Si la formation nécessite des prérequis correspondant à une ou plusieurs formations proposée par l'organisme.

¹⁹ Si la formation est facilitée par des prérequis correspondant à une ou plusieurs formations proposée par l'organisme.

Post-formation

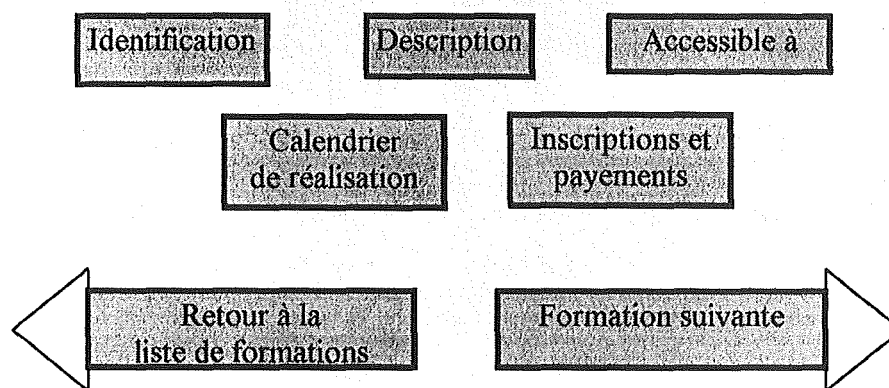
- **Titre** : <titre de la formation>
- **Type de formation** : <type de formation, mode de formation>
- **Durée** : <nombre de périodes, type de période>
- **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>

=====

- **Support de cours** : <type de support>
- **Certificat/diplôme**
 - * **Type** : <type d'entérinement²⁰>

Conditions d'obtention : <texte libre décrivant les conditions²¹>

- **Type de suivi associé à la formation** : <type de suivi>²²,
<commentaire suivi>



²⁰ A priori, les types d'entérinement retenus seraient : "aucun", "attestation de suivi de cours", "diplôme sous réserve d'évaluation", "diplôme sans évaluation".

²¹ Par exemple, pour un type d'entérinement de type "attestation de suivi de cours", la condition pourrait être "présence obligatoire à toutes les séances". Pour un type d'entérinement de type "diplôme sous réserve d'évaluation", les conditions pourraient préciser le mode d'évaluation et son coût éventuel, etc...

²² A priori les types de suivi retenus seraient "aucun", "à déterminer lors de la (des) séance(s)", "consultance auprès du formateur", "consultance auprès d'un expert du domaine". Le commentaire suivi permet de préciser les conditions auxquelles sont soumises ce suivi (coût, modalités,...).

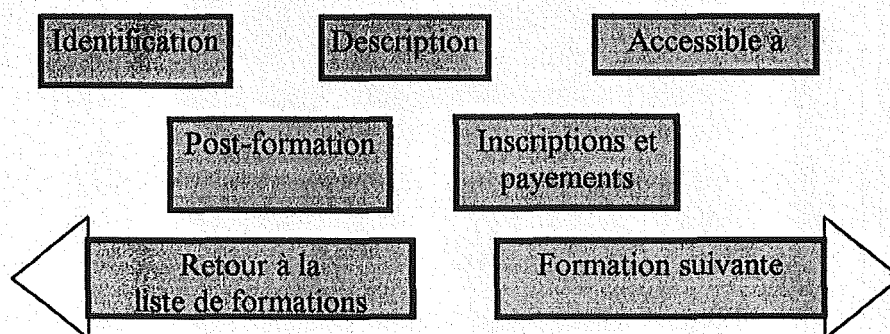
Calendrier de réalisation

- **Titre** : <titre de la formation>
- **Type de formation** : <type de formation, mode de formation>
- **Durée** : <nombre de périodes, type de période>
- **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>

=====

=

Sessions ²³	Langue	Lieu	Date(s)
<identif. session>	<code langue>	<Localité> < <u>Endroit</u> >	liste de {<date>, <horaire>}



²³ Une session correspond à une réalisation particulière de la formation. Elle peut s'étaler sur plusieurs jours (d'où une liste de dates).

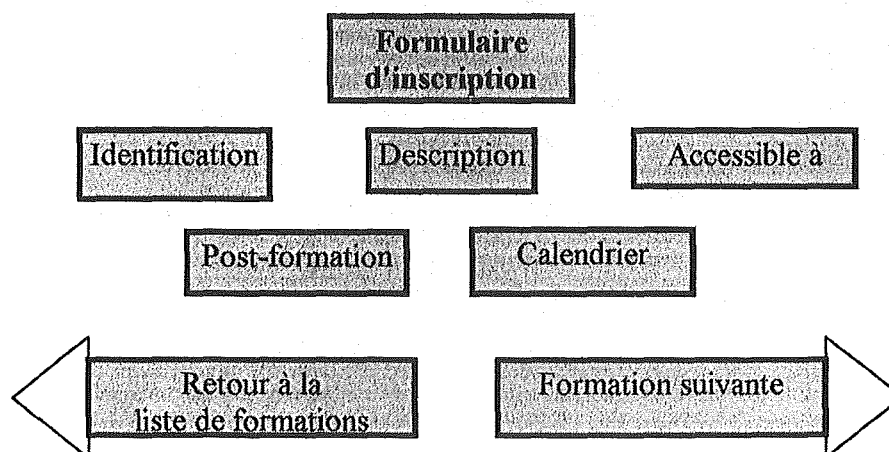
Inscriptions et paiements

- **Titre** : <titre de la formation>
- **Type de formation** : <type de formation, mode de formation>
- **Durée** : <nombre de périodes, type de période>
- **Organisme de formation** : <Nom de l'organisme, localité de l'organisme>

=====

=

- **Modalités d'inscriptions** : Vous pouvez vous inscrire électroniquement, par fax ou par la poste en utilisant le formulaire d'inscription ci-joint.
Votre inscription ne sera toutefois effective qu'après paiement.
- **Modalités de paiement** : Le montant de votre inscription peut être réglé²⁴
 - par envoi d'un chèque au nom de <nom de l'organisme, Localité>, accompagné d'une lettre rappelant l'objet de votre inscription.
 - par virement au compte <N° de compte> de <nom de l'organisme, Localité>, avec la mention "formation <N° de formation>/ référence de la session à laquelle vous êtes inscrit/nombre de participants".
- **Confirmation** : Une facture vous sera transmise dès réception de votre inscription.
 Une confirmation d'inscription vous sera transmise dès réception du paiement.
- **Annulation ou report d'une inscription**²⁵ : <texte libre décrivant les conditions correspondantes>



²⁴ A vérifier que ceci peut être standardisé.

²⁵ A vérifier que ceci peut être standardisé

Formulaire d'inscription

Responsable

- entreprise :
- Politesse :
- Nom :
- Prénom :
- Titre :
- Adresse :
- Code postal et localité:
- Téléphone:
- Fax:
- E- Mail:

Participant

- Politesse :
- Nom :
- Prénom :
- Titre :
- Adresse :
- Code postal et localité:
- Téléphone:
- Fax:
- E- Mail:
- Contact: Participant ☐ Responsable ☐

Envoyer

Recommencer

Cette exemple peut être défini de la manière suivante à l'aide de la grammaire BNF défini dans l'annexe 1. Des commentaires sont rajoutés au code. Ils sont délimités par '**'

```

Scenario formation ;
description scénario pour formation non-modulaire et inter ;
titre exemple scénario ;
** Début de la description des pages composant le scénario **
page formation_identification ;
    description page d'identification de la formation ;
    titre Identification ;
    ** Début de la description des sections de la page **
    section principale ;
        description section principale de la page identification ;
        alignement gauche ;
        ordre 1 ;
        ** Description de la liste composant la section **
        liste liste_identification ;
            description composant de la page identification ;
            type puce ;
            **description des éléments de la liste **
            texte item1 ;
                ** Première partie du texte **
                partie item1.1 ;
                contenu Titre : ;
                style gras ;
                ** Seconde partie du texte **
                partie item1.2 ;
                description titre de la formation ;
            texte item2 ;
                partie item2.1 ;
                contenu Réf : ;
                style gras ;
                partie item1.2 ;
                description identifiant de la formation ;
            texte item3 ;
                partie item3.1 ;
                contenu Type de formation : ;
                style gras ;
                partie item1.2 ;
                description type de formation, mode de formation ;
            texte item4 ;

```



```
    partie item4.1 ;
    contenu Durée : ;
    style gras ;
    partie item4.2 ;
    description nombre de périodes, type de période ;
texte item5 ;
    partie item5.1 ;
    contenu Langue : ;
    style gras ;
    partie item5.2 ;
    description liste des langues dans laquelle peut
        être donnée la formation ;
texte item6 ;
    partie item6.1 ;
    contenu Prix : ;
    style gras ;
    partie item6.2 ;
    description montant avec TVA ;
    partie item6.3 ;
    contenu TVAC (soit ;
    partie item6.4 ;
    description montant sans TVA ;
    partie item6.5 ;
    contenu HTVA) ;
    partie item6.6 ;
    description commentaires-prix ;
    partie item6.4 ;
    description réduction possible ;
texte item7 ;
    partie item7.1 ;
    contenu Organisme de formation : ;
    style gras ;
    partie item7.2 ;
    description nom de l'organisme ;
    destination ;
    partie itme7.3 ;
    description localité de l'organisme ;
texte item8 ;
    partie item8.1 ;
    contenu Cette formation fait partie du programme
    global : ;
    style gras ;
```

```
partie item8.2 ;
description Titre du programme ;
destination ;
```

**** Description de la page suivante du scénario ****

```
page formation_description
```

```
description page de description de la formation ;
titre Description ;
section principale ;
....
```

```
page formation_accessibilité ;
```

```
description page d'accessibilité à la formation ;
titre Accessible à ;
section principale ;
....
```

```
page formation_post ;
```

```
description page des post-formations ;
titre Post-formation ;
section principale ;
....
```

```
page formation_calendrier ;
```

```
description page du calendrier de réalisation de la
formation ;
titre Calendrier de réalisation ;
section principale ;
description section principale de la page du calendrier ;
alignement gauche ;
ordre 1 ;
liste liste_calendrier ;
description liste de la page calendrier ;
type puce ;
texte item1 ;
partie item1.1 ;
contenu Titre : ;
style gras ;
partie item1.2 ;
description titre de la formation ;
texte item2 ;
partie item2.1 ;
contenu Type de formation : ;
style gras ;
partie item2.2 ;
description type de formation, mode de formation ;
```

```

    texte item3 ;
        partie item3.1 ;
        contenu Durée : ;
        style gras ;
        partie item3.2 ;
        description nombre de périodes, type de période ;
    texte item4 ;
        partie item4.1 ;
        contenu Organisme de formation : ;
        style gras ;
        partie item4.2 ;
        description Nom de l'organisme ;
        destination ;
        partie item4.3 ;
        description localité de l'organisme;
alignement centre ;
ordre 2 ;
** Description du tableau composant la section **
tableau calendrier ;
    description tableau du calendrier ;
    nbr_colonne 4 ;
    ** description de la cellule 1,1 **
    numero_ligne 1 ;
    numero_colonne 1 ;
    texte cellule1.1 ;
        partie 1 ;
        contenu Sessions ;
    numero_ligne 1 ;
    numero_colonne 2 ;
    texte cellule1.2 ;
        partie 1 ;
        contenu Langue ;
    numero_ligne 1 ;
    numero_colonne 3 ;
    texte cellule1.3 ;
        partie 1 ;
        contenu Lieu ;
    numero_ligne 1 ;
    numero_colonne 4 ;
    texte cellule1.4 ;
        partie 1 ;
        contenu Date(s) ;

```

```

numero_ligne 2 ;
numero_colonne 1 ;
texte cellule2.1 ;
    partie 1 ;
    description identifiant Sessions ;
numero_ligne 2 ;
numero_colonne 2 ;
texte cellule2.2 ;
    partie 1 ;
    description code langue ;
numero_ligne 2 ;
numero_colonne 3 ;
texte cellule2.3 ;
    partie 1 ;
    description localite;
    partie 2 ;
    description endroit ;
    destination ;
numero_ligne 2 ;
numero_colonne 4 ;
texte cellule2.4 ;
    partie 1 ;
    description liste de date-horaire ;

```

```

page formation_inscription ;
    description page d inscription et de paiements pour la
        formation ;
titre Inscriptions et paiements ;
section principale ;
    description section principale de la page formation-
        inscription ;
alignement gauche ;
ordre 1 ;
liste liste_inscription ;
...

```

**** Description de la page contenant le formulaire ****

```

page formulaire_inscription ;
    description page formulaire pour l inscription à la
        formation ;
titre Formulaire d'inscription;
formulaire principale ;
    description du formulaire de la page;

```

```
alignement gauche ;
ordre 1 ;
texte titre 1 ;
    taille 12
    partie partiel ;
    contenu Representant ;
alignement gauche ;
ordre 2 ;
liste formulaire ;
    type puce ;
** Description des différents champs du formulaire **
editbox Entreprise ;
    libelle Entreprise ;
    long-max 80 ;
    long-champ 20
    type-donnee texte ;
editbox politesse ;
    libelle Politesse ;
    long-max 80 ;
    long-champ 20 ;
    type-donnee texte ;
...
** Description des boutons du formulaire **
submit ;
    methode post ;
    action reception.exe ;
    libelle Envoyer ;
Reset ;
    libelle recommencer ;
**Description de la navigation pour le scénario **
navigation;
** description des différents liens de la page formation_identification **
lien identification-description;
    source formation_identification ;
    destination formation_description ;
    position fin ;
    type bouton ;
    libelle Description;
lien identification-accessible;
    source formation_identification ;
    destination formation_accessibilité ;
    position fin ;
```

```
type bouton ;  
libelle Accessible à;
```

```
...
```

**** Description du lien vers formulaire_inscription à partir de la page : formation-inscription ****

```
lien inscription_formulaire  
    source formation_inscription ;  
    destination formulaire_inscription ;  
    position fin ;  
    type bouton ;  
    libelle formulaire d'inscription ;
```

IV. Rôle du Réalisateur

Le réalisateur est le dernier acteur à intervenir dans la création d'un scénario hypermédia. Comme le montre la Figure 4.14, le réalisateur dispose de types de scénarios mis à sa disposition par le scénariste pour réaliser son scénario. Il instancie ce scénario générique à l'aide de ressources multimédia et d'informations qui lui sont propres. La création de ressources multimédia n'est pas prise en compte dans cet outil. Il devra donc faire appel à des spécialistes pour créer des ressources ou bien les réaliser lui-même à l'aide d'outil graphique, de création de dessin, de création de vidéo, de séquenceurs musicaux, ...

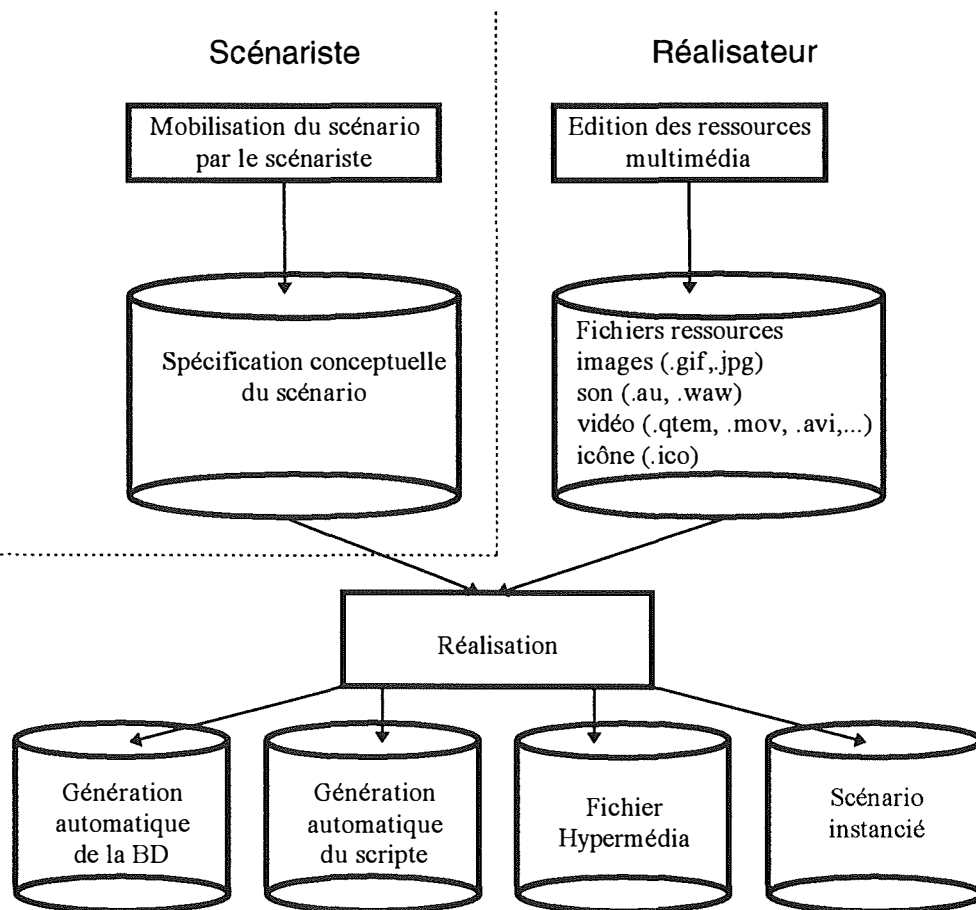


Figure 4.14: schéma descriptif de la réalisation

L'instanciation d'un scénario générique consiste à définir le contenu des objets. Il n'y a aucune emprise sur le positionnement de ceux-ci, de leurs alignements, ... L'ergonomie, la présentation générale est du ressort du scénariste.

Une fois le scénario instancié, l'outil générera toutes les pages hypermédia composant le scénario. Et si le scénario contient un ou plusieurs formulaires, un script correspondant à

chaque formulaire sera aussi généré. Chaque script aura pour but de manipuler les réponses reçues aux formulaires, d'en extraire les informations et de les enregistrer dans la base de données qui aura été créée à cet effet.

A. *Instantiation des scénarios*

Pour instancier un scénario, le réalisateur doit sélectionner un des scénarios définis par le scénariste. Pour cela, il aura la possibilité de parcourir la liste des types de scénarios existants et de leur description respective.

Une fois le scénario sélectionné, il devra donner toutes les informations nécessaires pour compléter l'ensemble des pages composant le scénario. Pour faciliter la représentation cognitive que le réalisateur se fait du type de scénario qu'il a sélectionné, l'outil lui proposera une visualisation globale des pages et des liens entre ces pages. Grâce à cela, il aura une vue générale et une idée plus précise de l'enchaînement des pages dans le scénario. Cette vue présentera d'une manière différente les pages qui sont complètement définies, des autres pages nécessitant l'intervention du réalisateur pour en définir les composants. Il aura la possibilité de voir la description de chacune de ces pages afin d'en sélectionner une page. Le réalisateur aura la possibilité d'instancier la page ainsi sélectionnée, dans le but d'en définir les objets et les liens la composant.

B. *Instantiation des pages*

A ce niveau, le réalisateur a sélectionné la page qu'il désire instancier. Une représentation de la page lui est proposée ainsi qu'une vue lui permettant de voir les liens intra-pages ainsi que les liens ou destination de cette page et les liens à partir de cette page.

Sur la représentation de la page, tous les objets la composant y sont représentés. Les objets nécessitant une instantiation de la part du réalisateur, sont représentés avec leur description de telle sorte que le réalisateur puisse connaître le but sémantique de l'objet. Pour les autres objets, ils apparaîtront dans leur forme terminale. Le réalisateur aura la possibilité de sélectionner l'objet qu'il désire instancier. Lors de la représentation d'une page contenant une section de type formulaire, il sera demandé au réalisateur de définir l'emplacement de la base de données et du script qui seront créés pour recevoir les réponses aux formulaires.

C. Instantiation des objets informationnels

Pour tous les objets informationnels, le réalisateur a la possibilité d'en définir le contenu ainsi que les caractéristiques à condition que celles-ci n'aient pas été définies par le scénariste.

Pour le TEXTE, il aura la possibilité d'en définir le contenu et d'en spécifier tous les attributs de style. Pour l'ICONE et pour l'IMAGE, il pourra indiquer quelle est la ressource multimédia pour cet objet. Pour les objets SON et VIDEO, il définira non seulement la ressource multimédia qui servira pour cet objet mais, en plus, il définira l'icône représentant cet objet, si celui-ci est de type déclenchement actionné. Pour les LISTES et TABLEAUX, ilinstanciera chaque ITEM ou chaque CELLULE, tout comme il instancierait un TEXTE.

D. Instantiation des objets interactifs

En ce qui concerne les objets EDITBOX et MULTIBOX, le réalisateur ne peut qu'en définir le LIBELLE étant donné que toutes les autres caractéristiques auront été définies par le scénariste. S'il le désire, il peut aussi décider d'une valeur initiale pour ces objets. Pour la LISTBOX et la COMBOBOX, il peut aussi lui donner un libellé et en définir la liste des ITEM si le scénariste ne l'a pas fait. Enfin, en ce qui concerne les TOGGLEBUTTON, il peut juste lui attribuer un libellé et en définir la liste d'ITEMS.

E. Instantiation des liens

Tous les liens définis par le scénariste n'ont pas obligatoirement un libellé défini. Si tel est le cas, le réalisateur aura la possibilité d'en définir un en se basant sur la description sémantique offerte par le scénariste.

Mais, dans la majorité des cas, le réalisateur ne devrait pas avoir à manipuler des liens. Il ne devrait pas en rajouter, ni en supprimer, car la navigation incombe au scénariste.

V. Outils

Grâce aux objets définis dans le schéma conceptuel présenté à la Figure 4.10, il est possible de spécifier la majorité des scénarios hypermédia. Mais il est clair que si l'on doit définir un nouveau scénario hypermédia en utilisant le langage de description, cela risque d'être relativement complexe et peu enrichissant vu que les objets manipulés sont d'un niveau relativement bas. Beaucoup d'éditeurs hypermédia sur le marché permettent d'ailleurs l'utilisation de ces mêmes objets de bas niveau et s'en contentent. Il serait donc plus intéressant, pour nous, de définir des objets de plus haut niveau. On pourrait, par exemple, rendre possible l'utilisation de structures de navigation de haut niveau. La création d'un nouveau scénario n'en sera que plus rapide et plus cohérente. Un outil d'utilisation de navigation structurée ou bien, un autre proposant des pages types, serait, par conséquent, bien utile. Ces outils se chargeraient eux-mêmes, de compléter le schéma de base (Figure 4.15).

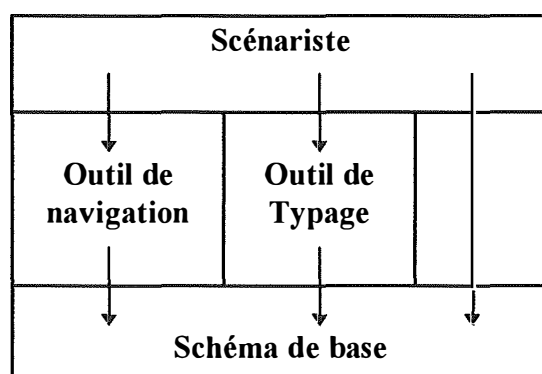


Figure 4.15: Outils du scénariste

A. *Outil de typage*

Cet outil a pour but d'aider à la création de nouvelles pages d'un scénario. Pour définir une nouvelle page, il faut en définir ses paramètres et tous les objets qui la composent. Et si on désire inclure dans le scénario, une nouvelle page qui a la même structure qu'une autre page précédente, il faudrait la redéfinir entièrement.

Il serait bien plus facile de pouvoir utiliser des modèles de page, contenant toutes les caractéristiques ainsi que tous les objets composant la page. C'est exactement à cela que sert cet outil.

Par exemple, si l'on désire créer un scénario contenant trois fois une page ayant les mêmes caractéristiques, il sera préférable de la définir une fois comme type de page et de l'utiliser trois fois plutôt que de la définir trois fois entièrement dans le même scénario.

Son utilisation dans le langage de description s'effectuerait de la manière comme définie dans l'annexe 2.

Pour chaque page, le scénariste doit en donner une description, un titre et a la possibilité de la composer entièrement ou bien d'utiliser un type de page prédéfini. S'il désire utiliser un type prédéfini, il lui suffit de spécifier le type de page qu'il souhaite parmi tous les modèles proposés par l'outil.

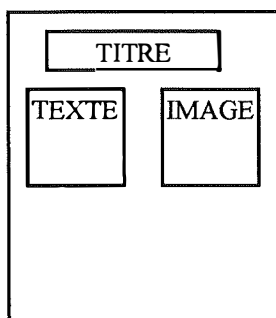
Cet outil permet donc d'utiliser des pages prédéfinies. Voici quelques types de pages prédéfinies qui pourront servir de modèle. Bien entendu, il devrait être tout à fait possible de définir d'autres modèles²⁶.

1. Page Présentation

Ce type de page permet de présenter un produit ou tout autre chose à l'aide d'un titre, d'une image, d'un texte et de son. Le type de page présentation EST DECOUPE PAR une SECTION de numéro d'ORDRE=1. Cette SECTION EST COMPOSE DE

- 1 objet TEXTE de numéro d'ORDRE=1 et aligné à *gauche*,
- 1 objet IMAGE, de même numéro d'ordre que le TEXTE, mais aligné à *droite*.
- 1 objet SON à déclenchement automatique.

a) *Représentation :*



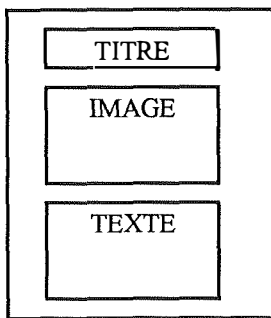
²⁶ Cette possibilité sera offerte ultérieurement

2. Page Couverture

Ce type de page sert de couverture à un livre ou tout autre chose, et elle comprend un titre, une image et un texte. Ce type de page EST DECOUPE PAR une SECTION de numéro d'ORDRE=1. Cette SECTION EST COMPOSE DE

- 1 objet TEXTE de numéro d'ORDRE=2 et aligné au *centre*,
- 1 objet IMAGE, de numéro d'ORDRE =1 et aligné au *centre*.

a) *Représentation :*



B. Outil de NAVIGATION

Les objets liens sont à la base de la navigation, mais on trouve dans les applications hypermédia des types généraux de navigation qui utilisent ces liens. Parmi les types généraux de navigation structurée, les plus fréquents sont : la navigation séquentielle, la navigation indexée, la navigation guidée, la navigation mixte, et la navigation globale.

Un outil permettant l'utilisation immédiate d'une structure de navigation est bien plus utile qu'un outil ne proposant que des pages et des liens. Il est bien plus facile de dire que l'on désire une navigation séquentielle entre les PAGES 1 à 4 que de définir le lien entre les pages 1 et 2, 2 et 3, 3 et 4. C'est pour cela que nous nous attardons à définir un tel outil qui permet d'utiliser des objets de navigation plus évolués que le lien. Ces objets de navigation peuvent être utilisés par le scénariste pour l'aider à créer la navigation dans son scénario.

1. Cadre conceptuel

Nous allons spécifier le schéma conceptuel des données (Figure 4.16) de l'outil de navigation dont disposera le scénariste. Ce schéma se base sur le concept de STRUCTURE pour définir la navigation. Ce concept de structure permet de définir un

Le scénariste définit la navigation à l'aide de liens, cependant, s'il le désire, il peut également utiliser les structures de navigation. Pour cela, il suffit au scénariste de spécifier le type de structure de navigation qu'il souhaite utiliser. Il lui donne un nom, définit le type de navigation à l'intérieur de cette structure, la position et le type de liens qui composeront la navigation dans cette structure. Enfin, il détermine la liste des pages ou des structures qui compose la structure. L'ordre de ces éléments de destination spécifié dans la liste, détermine l'ordre des composants qui constitueront la navigation. La première page déclarée correspondra à la page de départ de la navigation (la page source).

Les libellés de chaque liens peuvent être fixés par le scénariste. Pour cela, après la déclaration d'une page composant la structure, il peut définir un libellé qui servira pour tous les liens de cette structure ayant cette page comme destination. S'il ne souhaite pas fixer un libellé, il peut donner une description sémantique du lien.

Bien que le schéma conceptuel ne rende possible que l'utilisation d'objets de navigation pour une navigation inter-pages, il serait facile de le généraliser à la navigation intra-pages. Pour cela, il suffit d'ajouter une entité SECTION dans la Figure 4.16. jouant quasiment les mêmes rôles que l'entité PAGE Cette nouvelle entité aurait une association de COMPOSITION avec l'entité STRUCTURE, une association DIVISION avec l'entité PAGES, une association DESTINATION avec l'entité LIEN et une association APPARTIENT avec l'entité NAVIGATION. Cette entité SECTION n'a pas été représentée dans la Figure 4.16 pour des questions de lisibilité.

Cet outil propose les cinq types de navigation suivants :

2. Navigation séquentielle

Ce type de navigation permet de naviguer de page en page, en suivant une séquence (fil conducteur) prédéfinie. Cette structure permet juste de se rendre à la page suivante si le type de lien est *uni* (Figure 4.17), par contre, il permet de se rendre à la page suivante et de revenir à la page précédente si le type du lien est *bi* (Figure 4.18).

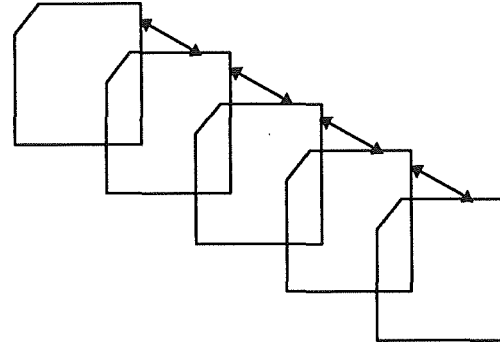
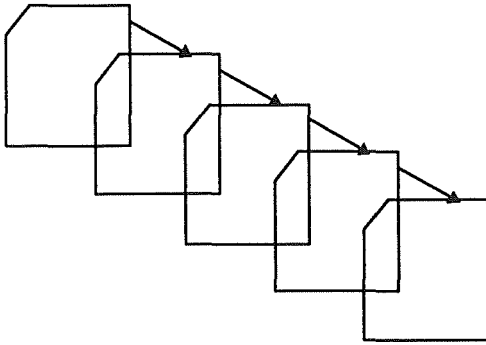


Figure 4.17: séquence uni-directionnelle **Figure 4.18: séquence bi-directionnelle**

3. Navigation indexée

Ce type de navigation fonctionne comme un index. A partir de la première page, on propose toutes les destinations possibles à l'utilisateur et il choisit la destination qui l'intéresse. Il ne lui sera pas possible de revenir à la page de départ si le type de lien est *uni* (Figure 4.19).

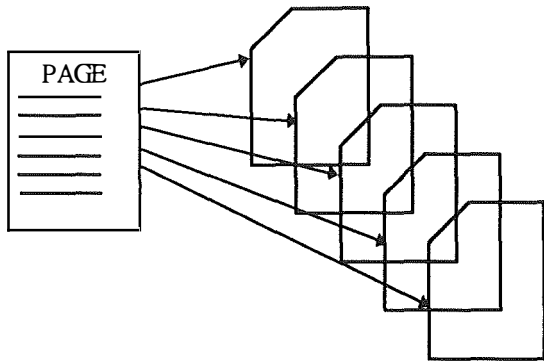


Figure 4.19: index uni-directionnel

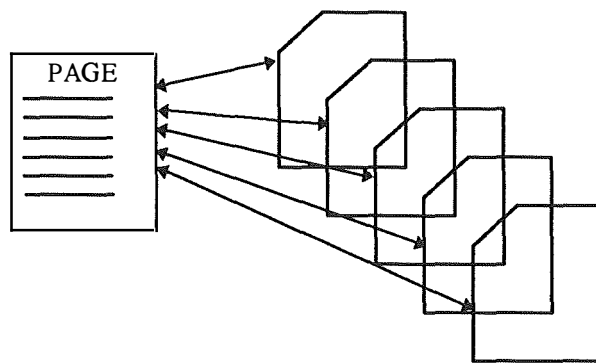


Figure 4.20: index bi-directionnel

Par contre, si le type de lien est *bi*, à partir de n'importe quelle page de la structure d'index, l'utilisateur aura la possibilité de revenir à la première page proposant l'index (Figure 4.20).

4. Navigation guidée

Ce type de navigation ressemble à un tour (voyage) guidé. On propose le départ du tour à l'utilisateur et il suit la navigation qui lui est proposée, et cela en suivant un chemin linéaire.

La première page présente le départ du tour guidé, en plus des objets définis sur cette page. A la fin du tour, l'utilisateur aura la possibilité de revenir au début du tour.

Si le type de lien est *uni*, il sera juste possible de se rendre d'une page vers la suivante (Figure 4.21). Ce type de navigation pourrait être comparée à une navigation séquentielle uni-directionnelle fermée. Par contre, si le type de navigation est *bi*, il sera possible de ce rendre, à partir d'une page du tour, vers la précédente ou vers la suivante (Figure 4.22).

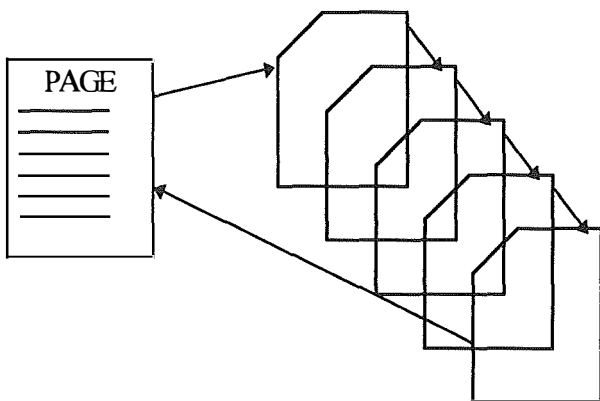


Figure 4.21: guide uni-directionnel

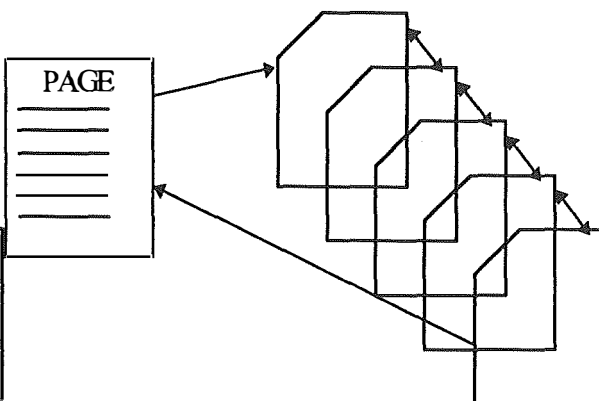


Figure 4.22: guide bi-directionnel

5. Navigation mixte

Ce type de navigation regroupe deux types précédents de navigation en un, à savoir, la navigation indexée et la navigation séquentielle. Si la navigation mixte est de type *uni*, alors elle comprend un index uni-directionnel et une séquence uni-directionnelle entre les pages accessibles à partir de l'index (Figure 4.23). Par contre, s'il est de type bi-directionnel, l'index ainsi que la séquence seront de type bi-directionnel (Figure 4.24).

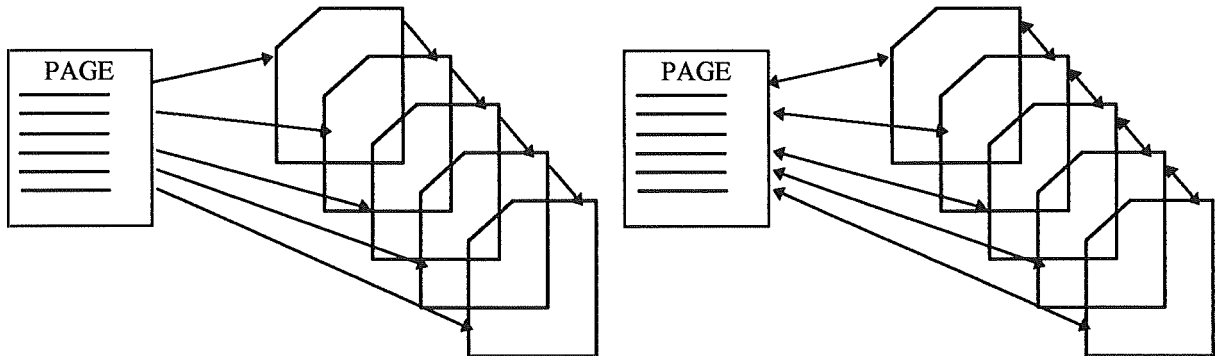


Figure 4.23: navigation mixte uni-directionnelle Figure 4.24: navigation mixte bi-directionnelle

6. Navigation globale

Ce type de navigation est certainement l'une des plus complètes, car à partir de n'importe quelle page, on peut se rendre directement vers toutes les autres pages de la structure.

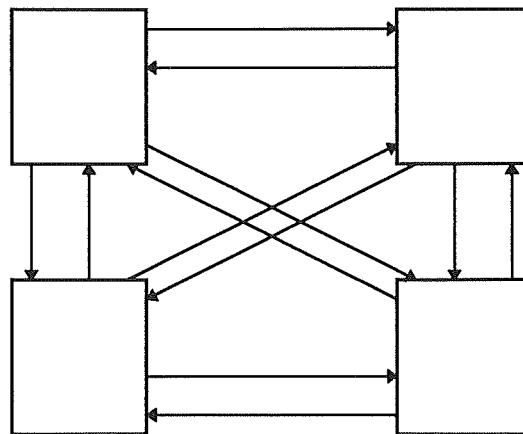


Figure 4.25: navigation globale

Evidemment, seuls les liens simples sont utiles avec ce type de navigation. Des liens doubles n'offriraient rien de plus qui n'est déjà offert par le type de navigation lui-même si ce n'est une répétition des liens existants (Figure 4.25).

7. Exemple

Grâce à cet outil de création de structure de navigation, le scénariste a la possibilité de développer des scénarios avec des navigations complexes sans s'occuper de tous les détails des différents liens composant cette navigation. Par exemple, s'il désire définir une navigation telle que représentée dans la Figure 4.26, il lui suffit de définir deux structures indépendantes en respectant l'ordre des pages : la première pour représenter la séquence A et une autre pour l'index B, ayant comme source la page A3. Par contre, si on avait voulu représenter une structure B incluse dans la navigation de la structure A, on aurait dû remplacer une des pages d'A par la structure B.

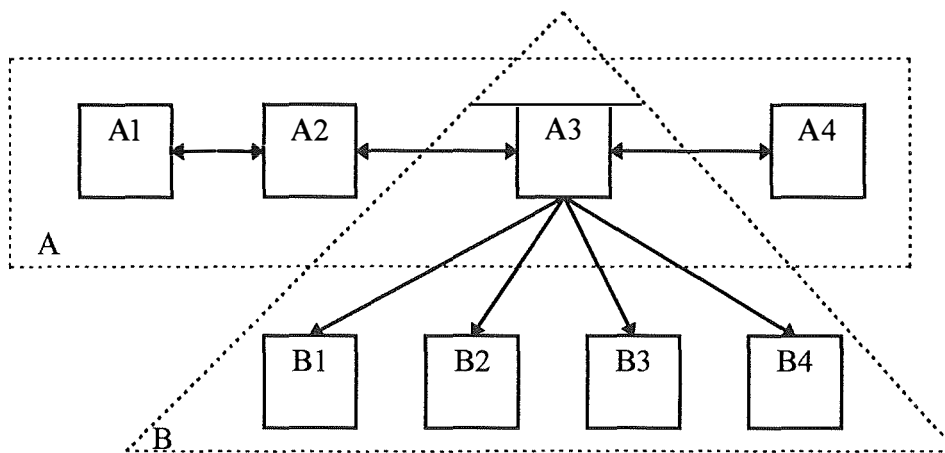


Figure 4.26: exemple de navigation composée

Cet exemple devrait être déclaré par le scénariste de la manière suivante :

...

Séquence A ;

```

navigation bi ;
position debut-fin ;
type texte ;
page A1 ;
libelle vers page A1
page A2 ;
libelle vers page A2
page A3 ;
libelle vers page A3
page A4 ;
libelle vers page A4
  
```

Index B ;

```

navigation uni ;
position fin ;
  
```

```
type texte ;  
page A3 ;  
libelle vers page A3  
page B1 ;  
libelle vers page B1  
page B2 ;  
libelle vers page B2  
page B3 ;  
libelle vers page B3  
page B4 ;  
libelle vers page B4
```

CHAPITRE 5

RÉALISATION

Ce chapitre est consacré à la mise en oeuvre des couches, des concepts et des outils définis dans le cadre théorique du chapitre 4 dans le cadre du projet SYRECOS.

I. Introduction

Après avoir défini les bases d'un langage de description de scénarios génériques et son instantiation, il est intéressant d'intégrer ce projet dans son contexte. En effet, le projet SYRECOS a pour objectif de créer un réseau interactif multimédia d'échanges de compétences comme support au développement régional.

Cette partie détaillera, d'une part, les besoins d'un point de vue technique ainsi que la réalisation technique du cadre théorique défini dans le chapitre 4.

Après avoir présenté l'environnement dans lequel on évoluera, l'intégration de l'outil dans le réseau urbain namurois, nous nous attarderons à présenter les outils destinés au scénariste ainsi que ceux destinés au réalisateur. Et à chaque fois, on expliquera exactement comment un acteur interagit avec son outil et cela de manière pratique.

II. L'environnement

Pour rappel, le but du projet SYRECOS est de concevoir et réaliser un télé-service régional dédié à l'échange de compétences. Il est donc clair que l'application devra être dédiée à un environnement distribué, accessible à distance, via un réseau.

Etant donné que l'un des objectifs est d'atteindre un maximum d'utilisateurs potentiels, ce projet sera basé sur la technologie Internet et privilégiera dans un premier temps la diffusion des formations proposées par les différents organismes. Ce choix d'un réseau basé sur la technologie Internet s'impose pour plusieurs raisons :

- utilisation de plus en plus répandue de ce réseau qui s'impose comme un standard de télécommunication informatique.
- accès possible à partir de toutes les plates-formes informatiques de manière transparente (système ouvert)
- existence de protocoles et d'outils devenus des standards dans le monde entier

Parmi les outils déjà existant pour INTERNET, il apparaît aussi évident que le choix de l'interface utilisateur doit reposer sur la technique du World Wide Web et ceci pour différentes raisons :

- il s'agit d'un système hypermédia ouvert et distribué
- il permet l'affichage de tout type d'information
- il possède des concepts simples et très facile à utiliser
- WWW est sans nulle doute l'application la plus répandue sur Internet
- son protocole de communication existe
- il est indépendant vis-à-vis de tout constructeur
- il existe des outils d'aide à la création et à la conception d'application WWW

En ce qui concerne les utilisateurs, ils devront pouvoir utiliser l'application en ayant simplement une connexion au réseau Internet et un outil de consultation du World Wide Web (un « browser » Cfr. Chapitre 3)

D'un point de vue technologique, le projet doit permettre une création rapide par des non-informaticiens , d'applications basées sur l'utilisation de serveur hypermédia World Wide Web. La réalisation d'application WEB requiert des compétences hétérogènes : scénariste, ergonomiste et informaticien. La réalisation requiert aussi la connaissance du langage HTML qui permet de développer les applications hypermédia pour le WEB.

Cette hétérogénéité représente un obstacle majeure pour l'utilisation de la technologie WEB pour les entreprises. Le cadre théorique définit dans le chapitre précédent (chapitre 4) permet de réduire cette hétérogénéité. En effet, grâce à ce langage, la création d'une application WEB se base sur un scénario générique de tel sorte qu'il suffit au réalisateur de l'instancier à l'aide d'outil graphique (WYSIWYG).

A. le réseau capillaire namurois

Le réseau capillaire namurois (Figure 5.27) est composé d'un serveur urbain et de plusieurs clients. Ces clients peuvent aussi bien être les organismes de formation, les clients à former que vous.

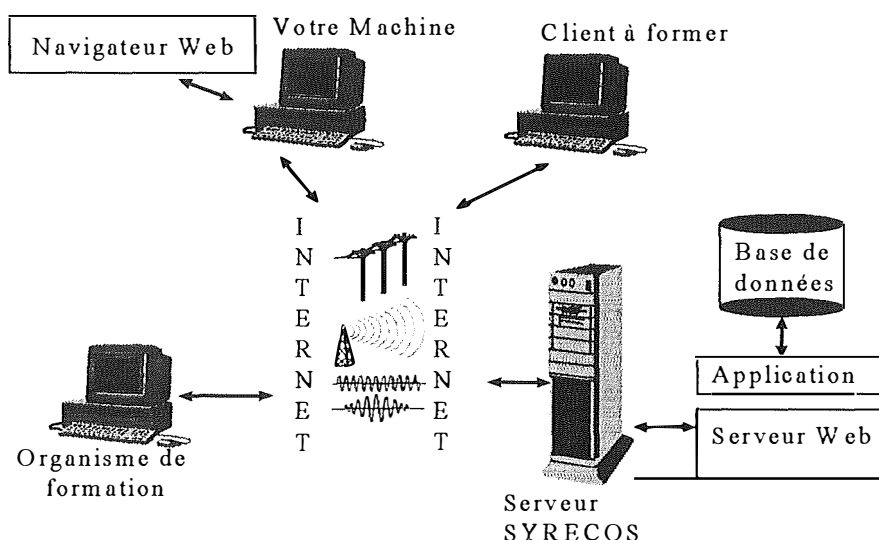


Figure 5.27: Architecture du réseau SYRECOS

Le serveur joue un rôle de centralisateur et de contrôleur. En effet, d'une part, il devra contenir une banque de données contenant toutes les formations offertes par les

différents organismes de formation et proposera aussi les scénarios génériques sur lesquels pourront se baser les organismes de formation pour développer leur application WEB. D'autre part, le serveur aura un rôle de contrôleur pour assurer la sécurité et la confidentialité des applications WEB.

C'est donc sur ce serveur que les scénarios d'application WEB devront être disponible aux différents réalisateurs potentiels du réseau Namurois. Ce sera aussi sur ce serveur que les scénarios instanciés devront être mis à disposition des autres utilisateurs du réseau capillaire namurois.

III. Implémentation

La plate-forme de développement sera constitué d'un micro-ordinateur PC et du système d'exploitation Windows 95.

Le World Wide Web peut être vu comme un ensemble de serveurs proposant des applications hypermédia (cfr. chapitre 3). Ces applications sont composées de pages définies à l'aide du langage HTML. Dès lors, les applications réalisées à l'aide du langage de description devraient être générées en HTML malgré que ce langage ne permet pas encore (de manière native²⁷) d'utiliser tous les concepts définis dans le chapitre 4. En effet, par exemple le déclenchement des sons et de la vidéo doivent se faire de manière explicite, c'est-à-dire par un click de l'utilisateur. Un déclenchement automatique n'est pas encore possible. De même, il n'est pas encore possible de définir des `tab_orders` entre des éléments.

Or, vu l'essor d'Internet et l'intérêt sans cesse grandissant que porte les sociétés informatiques à ce nouveau marché, beaucoup d'outils d'aide à la conception et à la génération de page HTML existent et sont de bonne qualité. Parmi les éditeurs de pages HTML les plus courants, on peut citer Webedit, Html Edit Pro, Word Internet Assistant, Live Markup, Webforms, ... Tout ces outils permettent de créer de nouvelles pages HTML mais aucun ne permet de se baser sur l'utilisation de modèles de pages, ni ne gère une application WEB complète et encore moins la navigation entre les différentes pages composant un scénario particulier.

Ces deux premières lacunes viennent d'être levées par un nouveau logiciel de Microsoft : FrontPage. En effet, FrontPage permet entre autre de gérer les pages composant une application WEB et d'en visualiser la navigation. Ce qui permet d'avoir une vue d'ensemble sur les relations des pages composant un WEB.

Cet outil peut être, dès lors, d'une très grande utilité pour le projet. Il se base sur un système hypermédia propre au langage HTML, permet au scénariste de définir de nouveaux types de pages et au réalisateur de les instancier. Il permet aussi au réalisateur

²⁷ HTML ne cesse d'évoluer de jour en jour. Il ne permet pas certain comportement de manière directe, mais il est possible d'en simuler certains en intégrant des plug-ins, des applets ou des Javascripts.

de créer une application WEB à partir d'un modèle de scénario et lui offre aussi la possibilité de visualiser graphiquement la navigation existante dans ce scénario à partir et à l'intérieur de chacune des pages le composant.

Par contre, bien que FrontPage permette de créer des applications WEB basées sur certains modèles d'application WEB génériques qu'il fournit, il ne supporte aucune fonctionnalité pour aider à la création de nouveaux modèles. De plus, aucun outil de navigation de haut niveau n'est offert tel que celui définit par notre outil de navigation du chapitre 4 à part un outil de table des matières.

Une partie des fonctionnalités des outils d'aide à la création de scénarios et à leur réalisation sera offerte par FrontPage de Microsoft. En ce qui concerne la définition de la navigation dans les scénarios (cfr. outil de navigation du chapitre 4), le scénariste aura à sa disposition l'outil SceneWeb que l'on a développé et qui est présenté ci-après.

A. FRONTPAGE

1. Présentation

FrontPage est un environnement client-serveur spécialement conçu pour les non-programmeurs, pour leur permettre de créer, diffuser et la gérer des applications WEB à l'aide d'outils intuitifs. Il rend inutile la connaissance du langage HTML. L'environnement de développement est WYSIWYG. Il supporte le langage HTML 3.0.

FrontPage est principalement composé de trois parties : un éditeur, un exploreur et un serveur.

1. L'éditeur de FrontPage permet de créer et d'éditer des pages WEB de manière WYSIWYG sans connaître HTML. Son utilisation est intuitive, dans la lignée des produits de la gamme Microsoft Office. Son utilisation est telle un traitement de texte. Sa base de travail est la page. La création de section s'effectue par l'utilisation de bookmarks.

Il permet de manipuler tous les objets informationnels de manière directe, tels que définit dans le chapitre 4, à l'exception de la vidéo et du son. En ce qui concerne la vidéo et le son, il faut utiliser la fonctionnalité de l'éditeur qui permet d'insérer des étiquettes HTML personnalisées. Il suffit donc d'utiliser cette fonctionnalité en insérant le code HTML souhaité.

Il supporte toutefois la création de formulaires et l'utilisation de tous les objets interactifs définis dans le chapitre 4 sans exception. On peut décider de l'action qui sera à effectuer lors de l'envoi d'un formulaire par un client WEB. FrontPage permet d'associer un script CGI personnalisé pour un formulaire ou bien d'y associer des scripts génériques pour traiter les réponses aux formulaires.

Des liens vers une autre section, une autre page ou même une page sur un autre site est possible. L'éditeur génère automatiquement le code HTML.

L'éditeur de FrontPage offre toute une série d'objet générique et immédiatement réutilisable. Par exemple, il offre un objet SEARCH qui aide la création d'une page servant de formulaire de demande de recherche. Il offre aussi la possibilité d'inclure d'autres pages HTML déjà existante dans la page que l'on développe.

Il permet aussi la création de pages types ainsi que leur réutilisation (outil de typage défini dans le chapitre 4). On peut donc définir de nouveaux types de page, spécifiques à un domaine et les réutiliser plusieurs fois pour des applications WEB différentes.

L'éditeur servira aussi bien au scénariste pour développer ses pages génériques qu'au réalisateur pour les instancier. Les commentaires d'un objet ou la description d'un objet que le réalisateur devrait introduire sur la page peut se faire à l'aide des annotations que l'éditeur permet d'insérer sur la page. Il s'agit là d'un excellent moyen pour que le scénariste puisse faire parvenir des informations au réalisateur. Une fois l'instanciation du scénario terminée, ces annotations ne seront pas visibles par les clients WEB.

2. L'explorateur donne une vue intuitive du scénario WEB complet. Il permet d'avoir une représentation hiérarchique des pages et il permet aussi d'avoir une vue graphique d'une page avec tous les liens entrants ou sortants de cette page. Cette fonctionnalité permettra au réalisateur de prendre connaissance de l'agencement général des pages composant le scénario WEB. L'explorateur permet aussi de gérer les applications WEB mises sur le réseau, à disposition des utilisateurs du WWW. Cet outil sera vraiment l'outil de base du réalisateur.

Il lui permettra de créer un nouveau site WEB à partir des scénarios génériques et d'appeler l'éditeur par un simple clique sur la page qu'il désire instancier. De cette manière, toutes les pages instanciées seront immédiatement mises à disposition sur le WWW sans aucune manipulation du réalisateur. Cet outil se base sur le contenu d'un fichier d'information qui est associé aux modèles de scénario. C'est ce fichier d'information qui lui indique les pages appartenant au scénario, la structure de fichier à adopter pour stocker les pages HTML dans le répertoire du serveur WEB.

3. Un serveur WEB est fourni dans l'environnement de FrontPage. Il s'agit d'un serveur WEB classique avec l'interface CGI pour l'exécution d'actions spécifiques à certains formulaires. Des outils d'administration et de contrôle sont aussi disponibles.

Toutefois, il est tout à fait possible d'utiliser d'autres WEB server que celui fourni par FrontPage. Mais si l'on désire utiliser toutes les fonctionnalités offertes par FrontPage, il faut installer les extensions de serveur que FrontPage fournit. Ces extensions existent pour la plupart des serveurs existant sur le marché.

2. Utilisation par le scénariste

Le rôle du scénariste ne diffère absolument pas du cadre général qui lui a été assigné. Seule la manière est différente car il utilisera FrontPage de Microsoft pour l'aider dans sa tâche de conception. A l'aide de ce logiciel, il pourra définir le contenu des pages qui composeront le scénario générique qu'il développe sans connaître un quelconque langage, ni même le langage de description du chapitre 4. Il composera son scénario à l'aide des objets visuels que lui propose FrontPage.

Tous les composants HTML lui sont proposés. Leur utilisation est simple et intuitive comme l'utilisation d'un traitement de texte. Parmi ces composants, il peut utiliser le TEXTE, l'IMAGE, l'ICONE, le SON, la LISTE, le TABLEAU, le LIEN, ... Tous ces objets sont conformes à leur définition dans le chapitre 4. Le logiciel propose aussi ce qu'il appelle des annotations. Ce dernier objet permet de positionner des commentaires dans la page. Ces commentaires sont uniquement visibles par ce logiciel. Il s'agit là d'un

excellent moyen de donner une description d'un objet que le réalisateur devra positionner.

Il est possible de définir des liens ayant comme source une image, une icône ou bien une partie d'un texte, tout comme cela est défini dans le langage de description. Ces liens peuvent avoir une destination interne à la même page (une section) , ou bien externe (une autre page). Un lien externe peut aussi bien être un lien vers une autre page du scénario générique ou une page quelque part sur le WEB qui sera déterminée en utilisant son URL.

En ce qui concerne la définition d'un formulaire, il doit disposer les objets interactifs et en définir les propriétés telles que le type de donnée, la longueur des champs, les items, ...

3. Utilisation par le réalisateur

Il est supposé ne posséder aucune connaissance spécifique pour ce qui est WWW, HTML,... Par contre, il est familiarisé avec le monde PC, il sait manipuler son environnement. Il est capable d'ouvrir les fichiers, d'en rechercher, d'utiliser et de comprendre des fenêtres,...

Pour instancier les objets d'une page, il lui suffit de suivre les commentaires que le scénariste lui a donné via les annotations.

Pour les objets informationnels, il peut définir les caractéristiques supplémentaires que le scénariste n'aurait pas définies. Par exemple, il définirait la liste des items d'une listbox que le scénariste n'aurait pas spécifier. Les items qu'il propose seront alors propre à son scénario.

B. SCENEWEB

1. Présentation

Cet outil est destiné à s'intégrer en amont de FrontPage. Il essaie de pallier aux principales lacunes de FrontPage. Il a pour but d'offrir les fonctionnalités nécessaires pour créer les bases d'un nouveau modèle d'application WEB. Ce modèle s'intégrera dans la liste des modèles proposés par l'explorateur de FrontPage de telle sorte qu'il soit accessible par le réalisateur lors la création d'une nouvelle application WEB.

En plus de cette première fonctionnalité, il offre aussi les fonctionnalités de création de navigation inter-pages. Son cadre théorique a été défini dans le chapitre 4 : '*Outil de navigation*'.

L'annexe 4 propose le schéma relationnel de la structure de la base de donnée utilisée par l'outil et l'annexe 5 propose le texte SQL complet de la création de la base de donnée.

Cet outil permet donc de créer un modèle de scénario WEB, composé des pages HTML et des objets de navigation générés à partir des structures de navigation définies. Les objets de navigation pouvant être inclus directement dans les pages même ou dans une frame. Les liens de ces objets de navigation peuvent être de type texte ou icône.

2. Utilisation par le scénariste

Le scénariste pourra utiliser l'outil SCENEWEB pour l'aider à créer un nouveau scénario et à en définir la navigation interne. Cet outil offre deux fonctionnalités distinctes. La première est de définir des scénarios et d'en sauvegarder la définition pour la compléter plus tard si besoin est. La seconde est la génération d'un scénario en page HTML et son intégration dans les modèles WEB proposés par FrontPage. Ces deux possibilités sont offertes au scénariste dans l'écran représenté à la Figure 5.28.

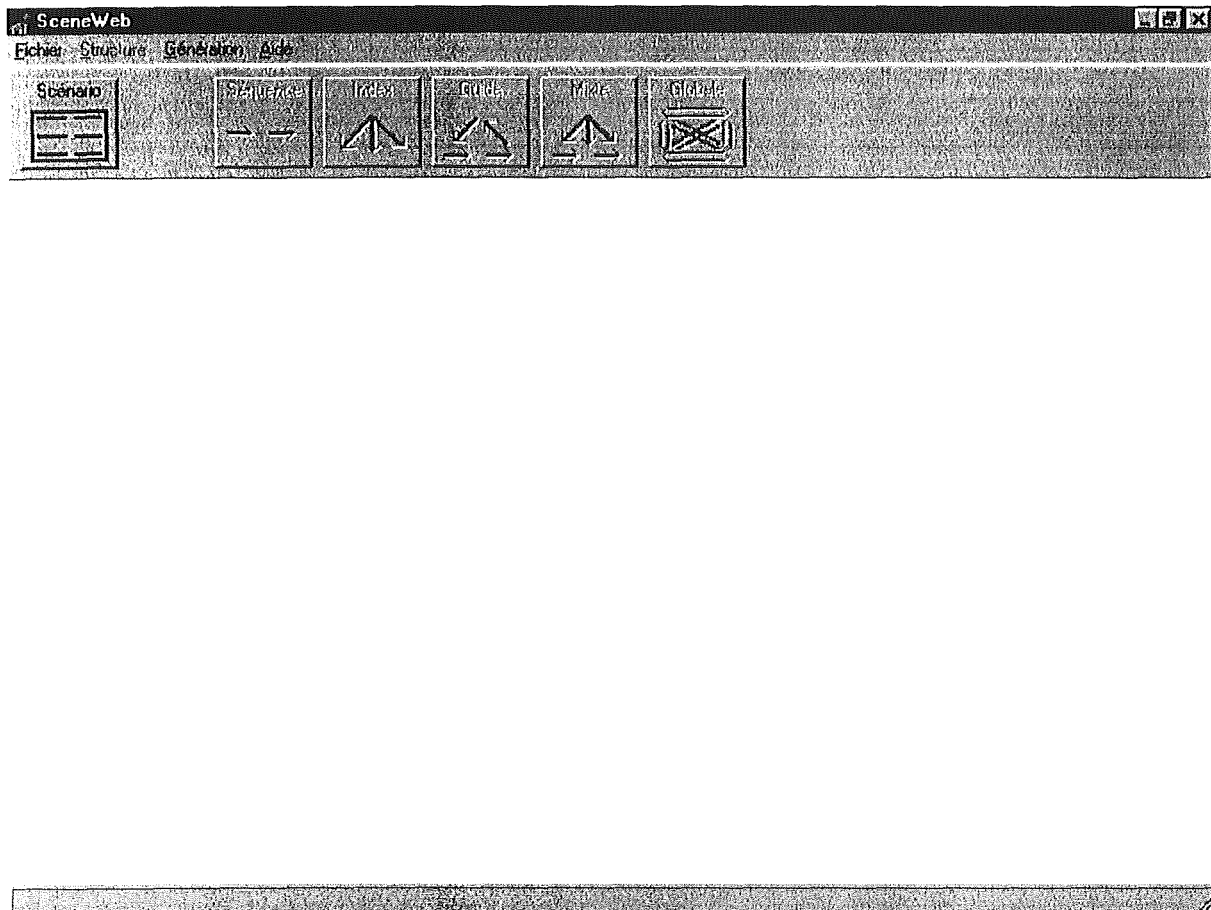
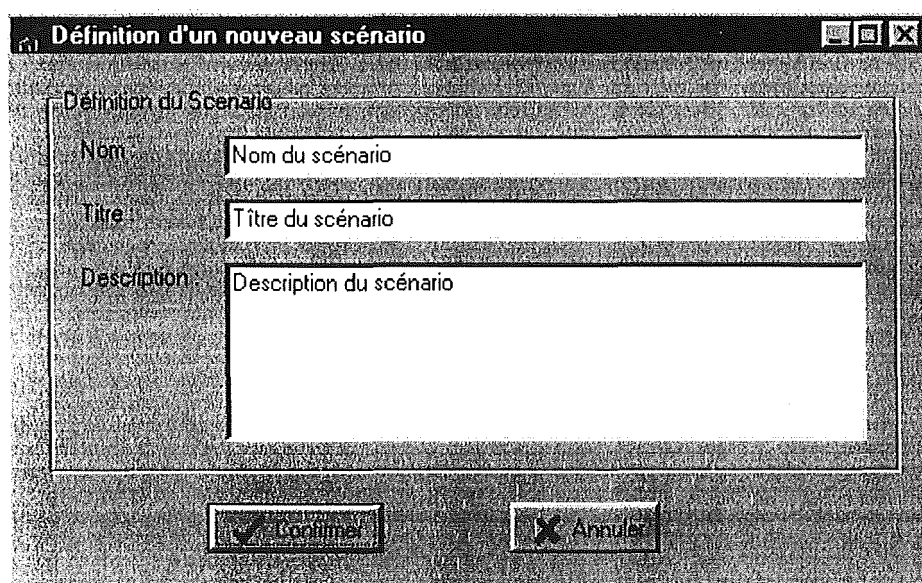


Figure 5.28: Ecran de départ de SceneWeb

a) Création d'un nouveau scénario

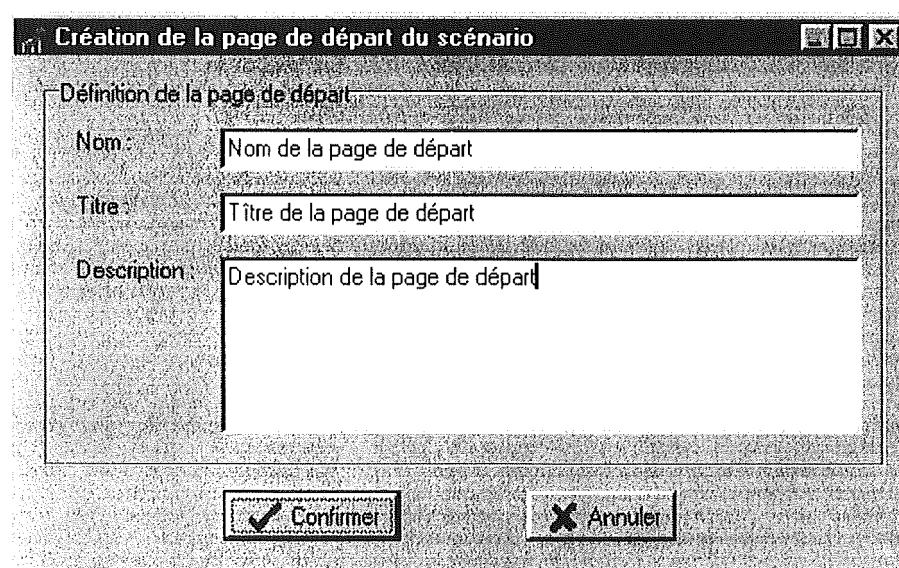
Si le scénariste a opté pour la définition d'un nouveau scénario, l'application lui demande alors de donner un nom, un titre et une description (facultative) au nouveau scénario (Figure 5.29). Le nom servira juste à identifier le scénario alors que le titre apparaîtra dans la liste des scénarios disponibles pour le réalisateur en lui donnant en plus la description de ce scénario.



The screenshot shows a Windows-style dialog box titled "Définition d'un nouveau scénario". It contains three input fields: "Nom" with the placeholder text "Nom du scénario", "Titre" with "Titre du scénario", and "Description" with "Description du scénario". The "Description" field is a larger text area. At the bottom, there are two buttons: "Confirmer" (with a checkmark icon) and "Annuler" (with an 'X' icon).

Figure 5.29: Définition d'un nouveau scénario

Une fois la définition du scénario encodée, le scénariste doit aussi définir une nouvelle page qui deviendra la page de départ du scénario. La définition d'une page de départ est obligatoire pour tous les scénarios. Pour cela, il définit un nom, un titre et une description qui est elle, facultative (Figure 5.30).



The screenshot shows a Windows-style dialog box titled "Création de la page de départ du scénario". It contains three input fields: "Nom" with the placeholder text "Nom de la page de départ", "Titre" with "Titre de la page de départ", and "Description" with "Description de la page de départ". The "Description" field is a larger text area. At the bottom, there are two buttons: "Confirmer" (with a checkmark icon) and "Annuler" (with an 'X' icon).

Figure 5.30: Définition de la page de départ d'un nouveau scénario

Une fois les éléments caractéristiques d'un nouveau scénario validés, le scénariste a la possibilité de définir des nouvelles structures de navigation dans son scénario (Figure 5.31). Il a le choix entre les types de structures suivantes : Séquence, Index, Guide, Mixte et Globale. Ces structures de navigation sont celles définies dans le langage de

description. Cet écran est disponible uniquement lorsqu'un scénario est ouvert ou en cours de définition.

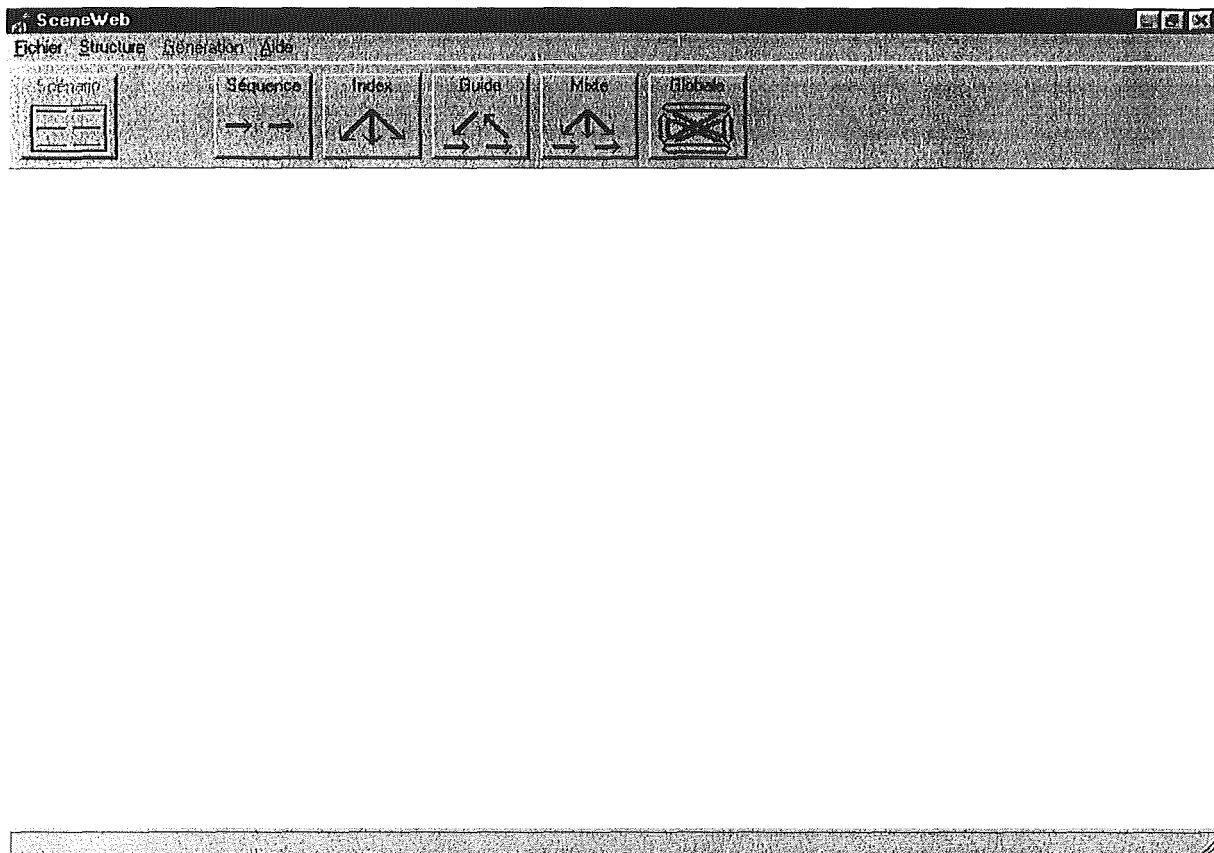


Figure 5.31: Ecran d'ajout de structure à un scénario

Pour définir une structure, il devra sélectionner le type de structure qu'il désire, lui donner un nom, un titre et une description (facultative). L'écran permettant cette définition est représentée à la Figure 5.32.

The image shows a dialog box titled 'définition d'une structure'. Inside the dialog, there is a section labeled 'Structure sequence'. Below this label, there are three text input fields: 'Nom :', 'Titre :', and 'Description :'. Each field contains a placeholder text: 'Nom d'une structure', 'Titre d'une structure', and 'Description d'une structure' respectively. At the bottom of the dialog, there are two buttons: 'Confirmer' (with a checkmark icon) and 'Annuler' (with an 'X' icon).

Figure 5.32: Définition d'une structure de type séquence

Après avoir défini la structure, il devra en donner les propriétés (Figure 5.33).

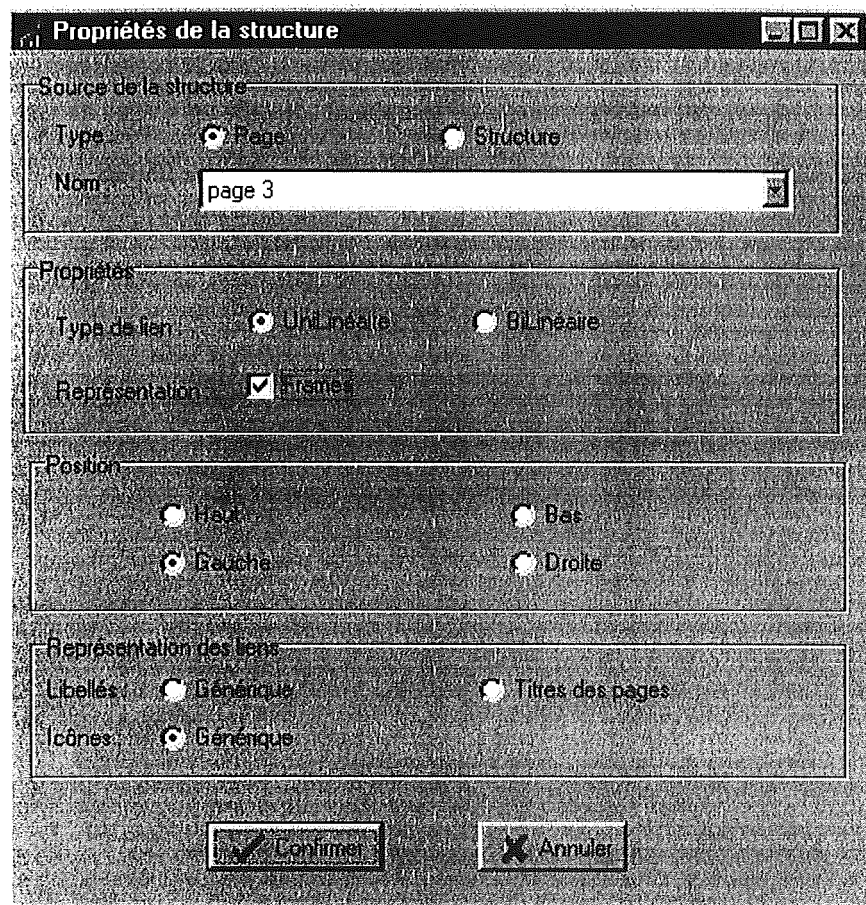


Figure 5.33: Définition des propriétés d'une structure

1. La source de la structure : c'est-à-dire à partir de quelle page ou quelle autre structure cette nouvelle structure sera disponible.
2. Le type de lien : unilinéaire ou bilinéaire
3. La représentation de l'objet de navigation dans une Frame ou pas. La Figure 5.33. représente le choix d'une représentation en Frame.
4. La position de l'objet de navigation : Haut, bas, gauche, droite dans le cas d'une Frame ; début, fin et début-fin dans le cas contraire. La Figure 5.33. représente les choix possibles de position dans le cas d'une représentation en Frame.
5. La représentation des liens : il a le choix entre des liens de types texte ou Icône. Dans le cas d'un type texte, le libellé peut-être soit les titres des pages (valable pour tous les types de structures) ou un libellé générique (valable pour tous les types de structure sauf la globale). Dans le cas d'icône, seules des icônes

génériques sont proposés. En effet, il serait idiot pour un outil de haut niveau de demander au scénariste l'icône qu'il désire utiliser pour tel lien, et cela pour tous les liens de la structure. On perdrait le caractère générique de l'outil.

Le scénariste aura ensuite à définir la liste des pages appartenant à la structure dans l'ordre de leur agencement dans la structure Figure 5.34. Pour cela, il pourra soit définir de nouvelles pages (Figure 5.35) ou bien réutiliser des pages déjà existantes (Figure 5.36).

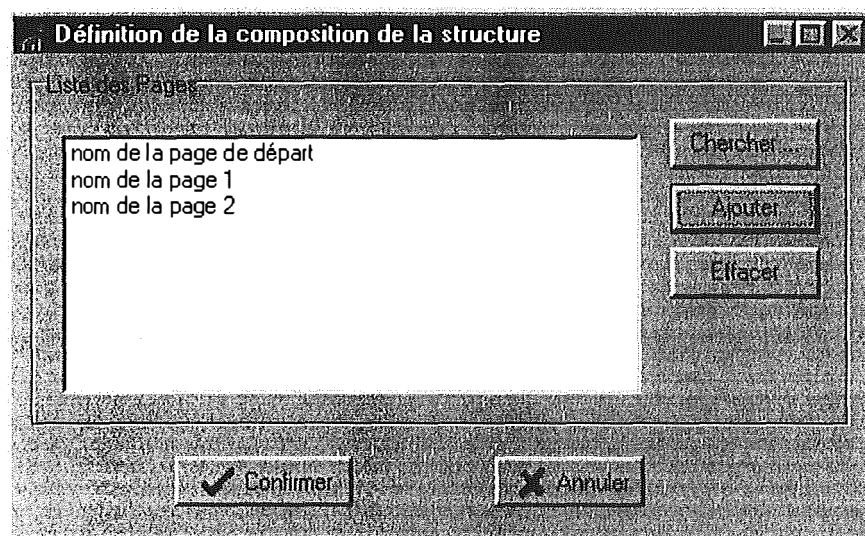


Figure 5.34: Liste des pages de la structure

Pour chacune des pages de la structure, il devra lui donner un nom, un titre et une description. Le titre servira de titre de page pour les clients WEB et la description permettra au réalisateur de connaître le but rechercher par la création de cette page.

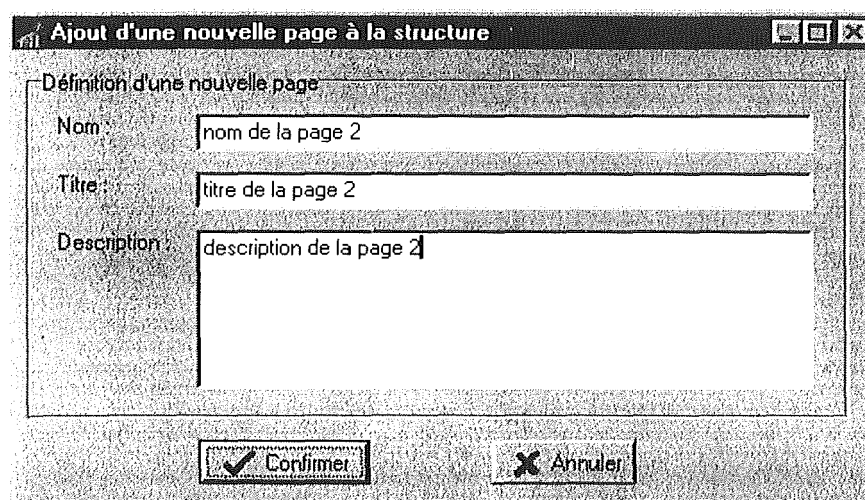


Figure 5.35: Définition d'une nouvelle page

Le fait de réutiliser des pages déjà existantes permet de définir une navigation hypertextuelle plus riche et plus complexe que les structures de navigation traditionnelles.

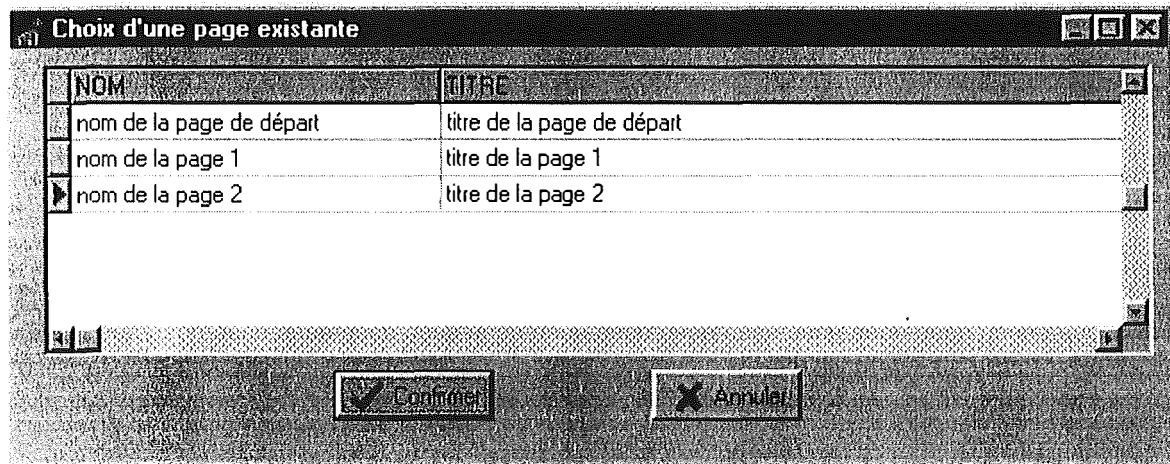


Figure 5.36: Choix d'une page déjà existante

Une fois la définition d'un nouveau scénario et des structures faite, le scénariste n'a plus qu'à enregistrer ces définitions.

b) Ouverture d'un scénario déjà existant

SceneWeb permet aussi au scénariste de compléter un scénario déjà existant. Pour cela, il lui suffit de sélectionner le scénario qu'il désire compléter (Figure 5.37) et ensuite de définir de nouvelles structures. Pour cela, il se retrouvera à l'écran représenté à la Figure 5.31. il lui suffit d'opérer de la même manière qu'il le ferait pour créer les structures d'un nouveau scénario.

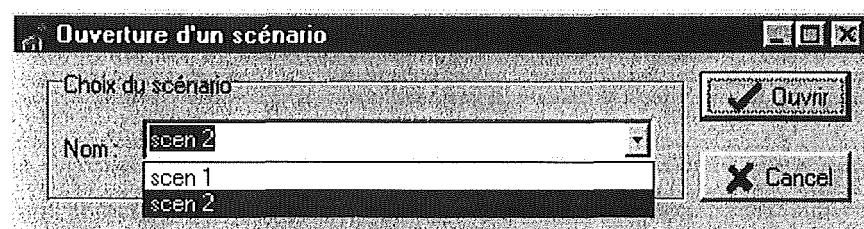


Figure 5.37: Ouverture d'un scénario déjà existant

c) Génération d'un scénario

Le scénariste peut générer un nouveau modèle d'application WEB à partir de scénario défini. Pour cela, il lui suffit de sélectionner le scénario qu'il désire générer parmi la liste des scénarios définis (Figure 5.38). Une fois la sélection confirmée, le scénario sera automatiquement généré et intégré dans les 'WEB templates' de FrontPage. Le fichier

d'information du 'WEB template' nécessaire pour FRONTPAGE sera aussi généré. Les pages HTML et les objets de navigation seront créés. Le scénariste aura encore la charge de définir les objets composants les pages à l'aide de l'éditeur de FrontPage afin que le scénario complet soit disponible pour le réalisateur.

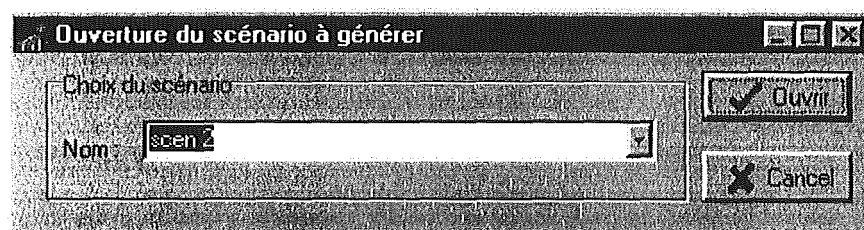


Figure 5.38: Sélection du scénario à générer

3. Perspectives et améliorations futures

Ce logiciel pourrait être l'oeuvre d'améliorations. La principale amélioration que l'on pourrait lui apporter serait de le rendre visuel. Par le rendre visuel, il faut entendre avoir une représentation graphique du scénario.

Cette représentation permettrait t de voir l'organisation des structures à l'intérieur composant le scénario. Cela permettrait au scénariste d'avoir une vue globale des structures qu'il crée et de la navigation à l'intérieur de ce scénario. Cela permettrait donc de visualiser les liens qui existent entre les différentes structures.

Un double-click sur une structure permettrait au scénariste de « rentrer » à l'intérieur de cette structure pour en visualiser la composition, l'agencement des pages, la navigation entre ces pages. Il devrait pouvoir déplacer les pages à l'intérieur de cette structure ce qui en modifierait l'ordre des pages de cette structure, ainsi que les liens composant la navigation entre les pages.

Pour que cet outil soit vraiment visuel, non seulement il devrait être possible de déplacer les structures d'un scénario (ainsi que les pages d'une structure, comme mentionner ci-dessus), mais il devrait aussi être possible de voir les propriétés d'une structure et de les modifier. Pour cela, il serait utile d'avoir une fenêtre du type « structure inspector ». Cette fenêtre afficherait les propriétés de la structure active.

IV. Cas pratique

Nous allons, à titre d'exemple, suivre tout le développement, par le scénariste, de l'exemple de scénario générique présenté dans le chapitre 4. Cet exemple est composé de sept pages. Ces sept pages peuvent être regroupées en 2 structures de navigation (Figure 5.39). La première structure est composée de 6 pages. Il s'agit d'une structure de navigation globale. La seconde structure est une structure de type séquentiel, unilinéaire entre la page 'Inscription et paiement' et 'Formulaire d'inscription'.

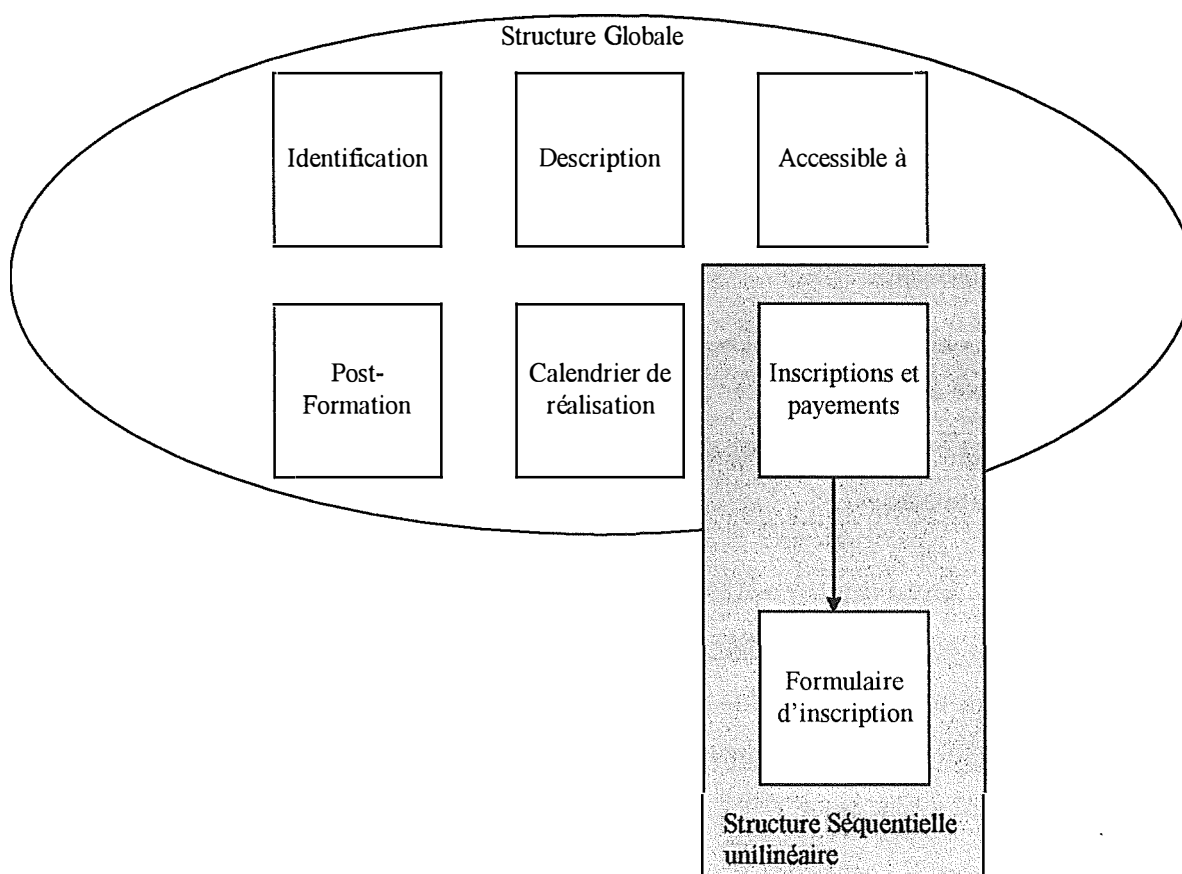


Figure 5.39: Structure de navigation de l'exemple du chapitre 4

Le scénariste peut donc utiliser l'outil SceneWeb pour créer les bases de ce scénario. Il lui suffit de créer un nouveau scénario, ayant comme page de départ la page 'Identification' et composé de deux structures. La première est une structure de type globale et ayant comme source la page 'Identification'. A cette structure, il lui rajoute les cinq autres pages composant cette structure. Cette structure doit avoir les propriétés suivantes pour respecter l'exemple :

- option Frame désactivée
- position : fin
- représentation des liens : libellés - titres des pages

En ce qui concerne la seconde structure, le scénariste doit choisir une structure de type séquence. La source de cette structure est la page 'Inscription et paiements'. Les propriétés de cette structure sont :

- type de lien : Unilinéaire
- option frame désactivée
- Position : fin
- représentation des liens : libellés - titres des pages

Il reste au scénariste à sauvegarder ce nouveau scénario et de le générer. La génération transformera ces structures en templates de pages pour FrontPage. Il suffira au scénariste d'exécuter l'éditeur de FrontPage et de compléter chacune de ces pages avec les objets qui les composent.

Pour cela, il se rend dans le répertoire WEBS de FrontPage. Il sélectionne le sous-répertoire correspond au nom du scénario qu'il a défini. Et pour chacune des pages composant ce répertoire, il les ouvre pour les éditer. La Figure 5.40 illustre l'éditeur de FrontPage lorsque le scénariste a sélectionné la page 'Identification' généré par SceneWeb. La Figure 5.41 illustre la même page après édition de la part du scénariste.

Une fois toutes les pages du scénario éditées, le scénariste à terminer son rôle. Le scénario générique est dès lors terminer et prêt à l'emploi. SceneWeb s'étant occupé de l'intégration avec FrontPage. Il reste alors au réalisateur de continuer pour instancier ce scénario.

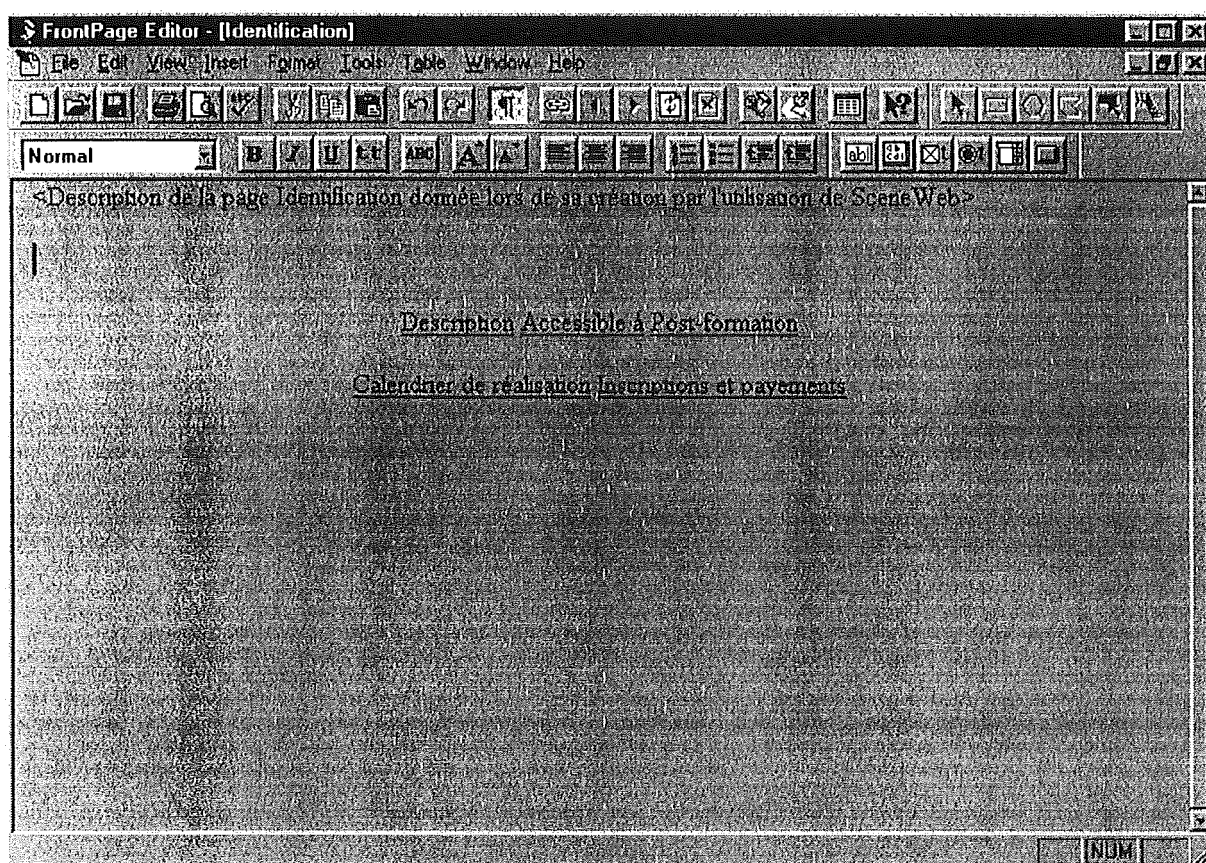


Figure 5.40: Page Identification après génération de SceneWeb, avant édition

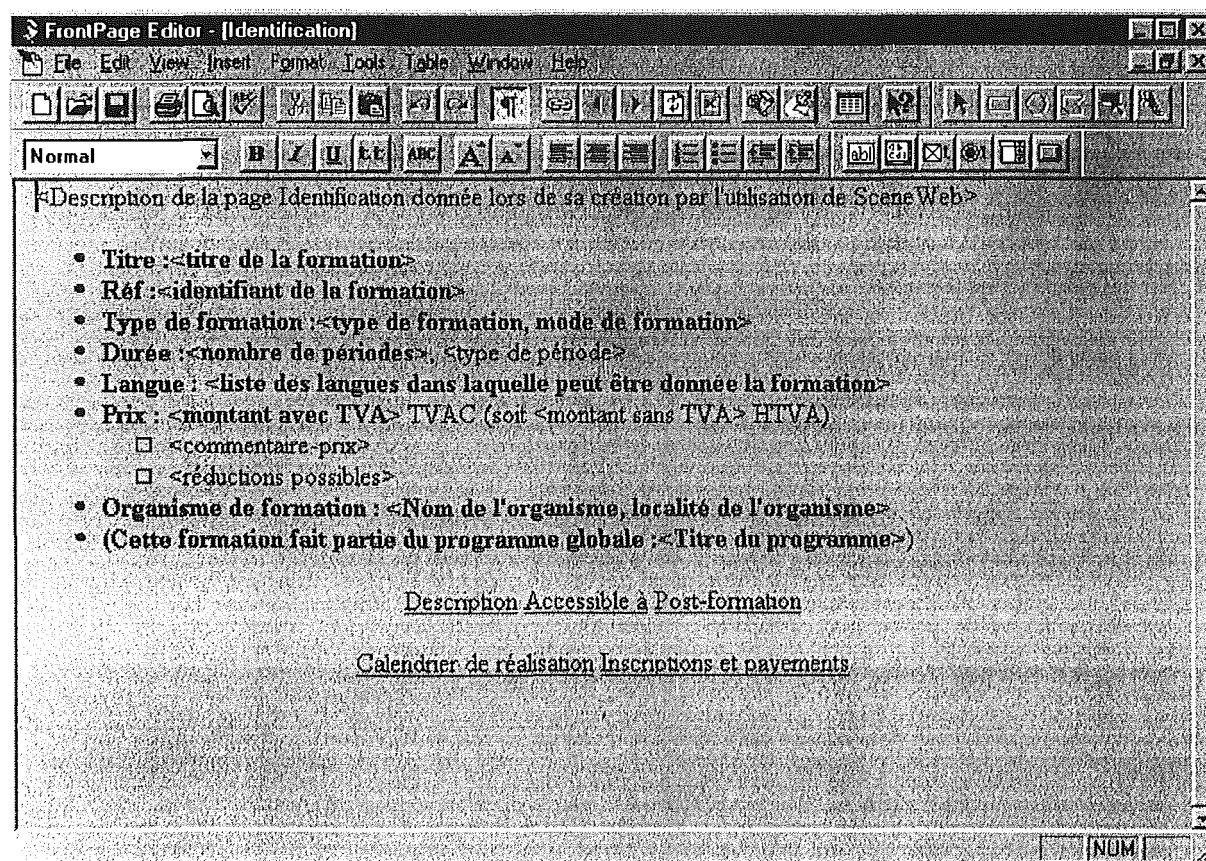


Figure 5.41: Définition de la page identification

Le rôle du réalisateur commence par la sélection du scénario générique qu'il désire instancier (Figure 5.42). Avant de confirmer sa sélection, il peut visualiser la description du scénario qu'il désire.

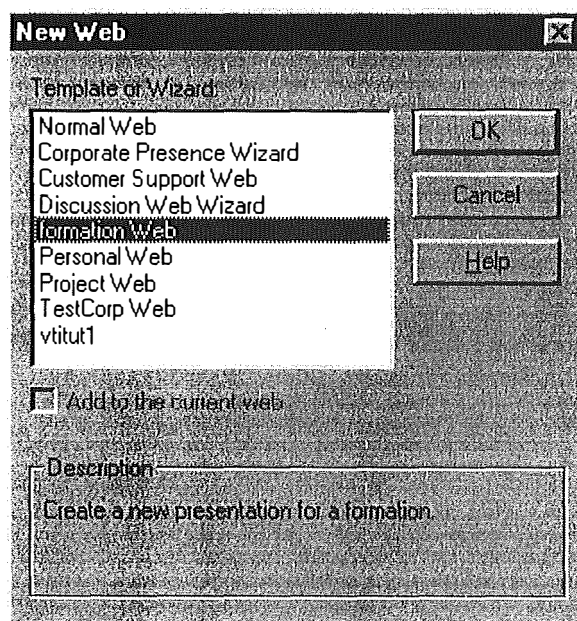


Figure 5.42: choix des différents scénarios génériques dans FrontPage explorer

Une fois le scénario sélectionné, l'explorateur de FrontPage lui propose une vue générale du scénario (Figure 5.43). Ce qui permet au réalisateur de prendre connaissance avec la structure de la navigation pour ce scénario. En sélectionnant une page dans la 'Outline View', il peut visualiser toutes les pages ayant comme destination cette page, ainsi que toutes les pages destinations de celle-ci : 'Link View'.

Il a la possibilité de sélectionner la page qu'il désire instancier en double-cliquant dessus. A ce moment, il rentre dans l'éditeur, avec la possibilité d'éditer la page à partir de l'état où le scénariste l'a laissée (Figure 5.41). Il ne lui reste, à ce moment, qu'à remplacer les annotations du scénariste par les valeurs qui sont propres à sa formation (Figure 5.44). Une fois qu'il a instancié toutes les pages de sa formation, la nouvelle application Web est mise à disposition du réseau INTERNET, et tout les clients potentiels peuvent y accéder. Les personnes intéressées pourront suivre la présentation de la formation à partir de leur client WEB et suivre les liens pour accéder aux autres pages. La Figure 5.45 illustre la page identification instanciée telle que pourrait la voir un client Netscape.

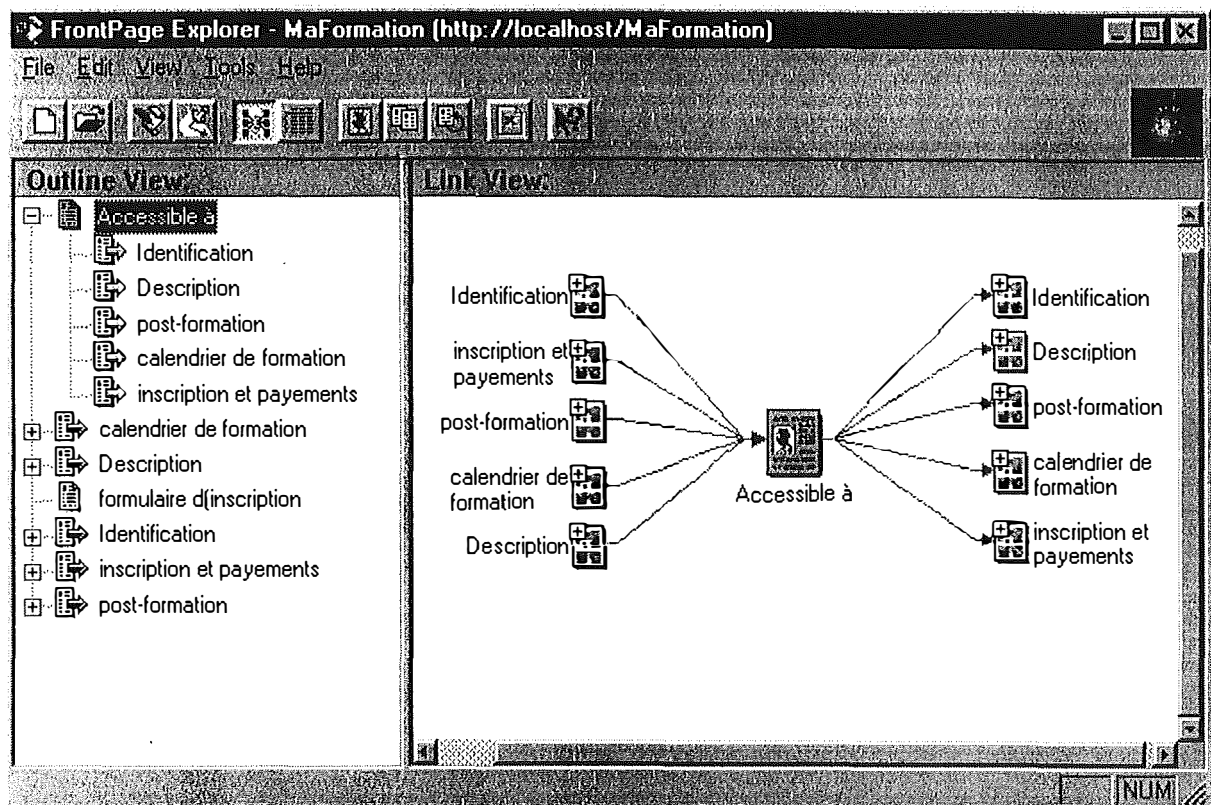


Figure 5.43: Scénario vu par l'explorer de FrontPage

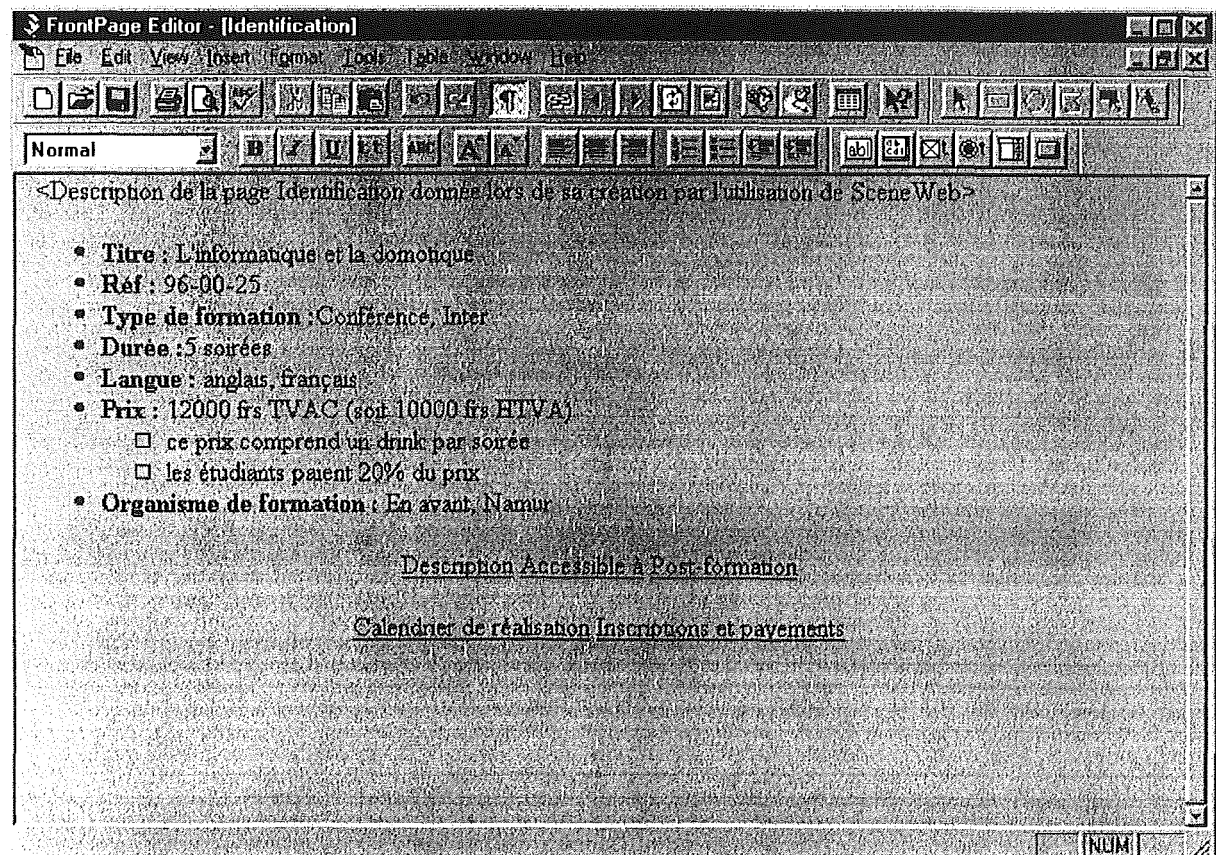


Figure 5.44: Page Identification instanciée par le réalisateur

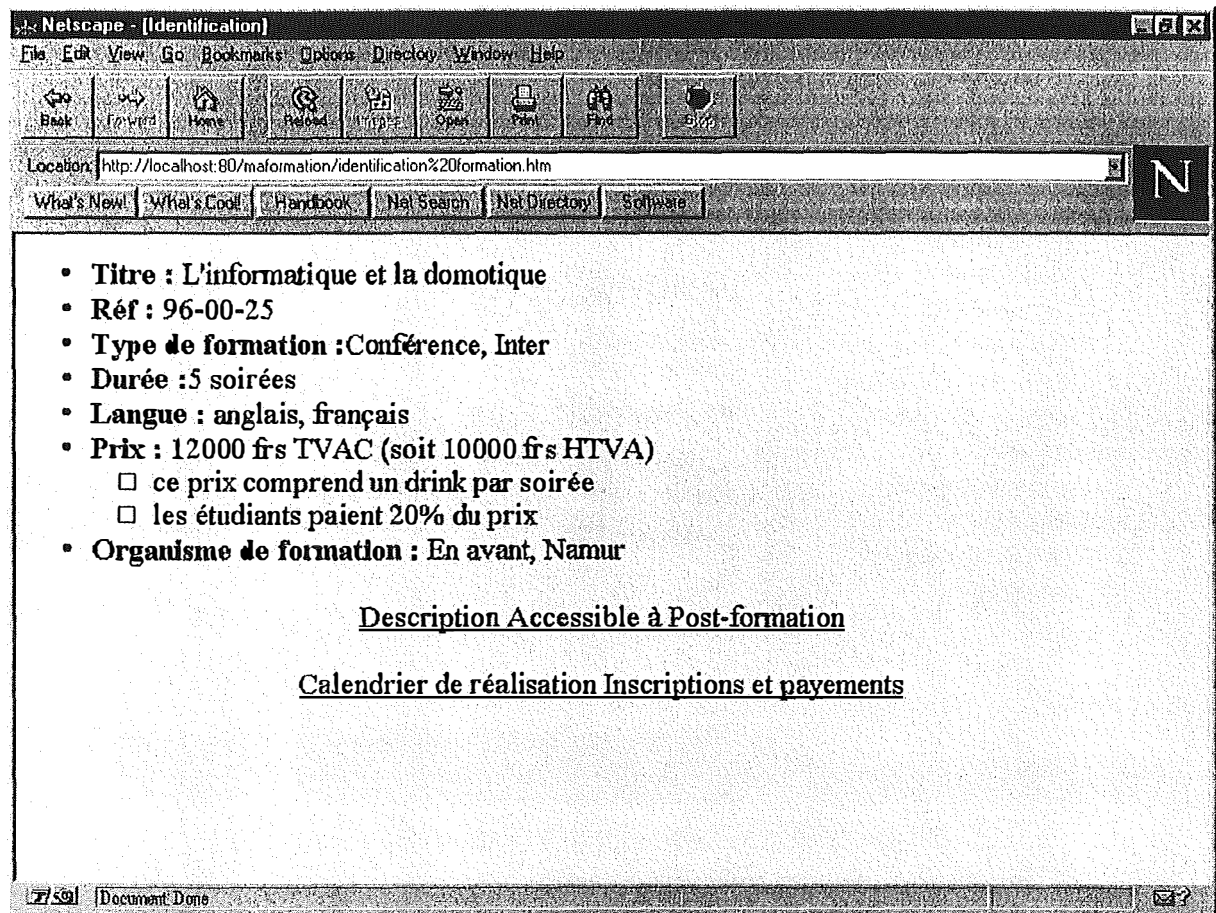


Figure 5.45: Visualisation de la page Identification par un client Netscape

CHAPITRE 6

PERSPECTIVES MÉTHODOLOGIQUES ET CONCLUSION

Ce chapitre présente une perspective méthodologique pour développer des applications hypertexte ainsi que les conclusions et perspectives d'avenir de ce travail.

I. Perspectives méthodologiques

Nous allons présenter brièvement deux méthodes orientées vers le développement d'interfaces. La première méthode est RMM et la seconde est celle proposée par l'équipe du projet TRIDENT. Après leur présentation, nous dégagerons les principaux avantages et inconvénients que ces méthodes ont dans le cadre d'un développement d'applications hypertexte. Nous essayerons enfin de dégager les lignes directrices que devrait prendre une méthode plus adaptée à la modélisation d'applications hypertexte suivant des règles ergonomiques dédiées à l'interface homme-machien (IHM)..

A. Relationship Management Methodology (RMM)

Cette méthodologie a été développée par Izakawitch [Izakowitz, 1995], et est dédiée à la conception et à la construction d'applications hypermédia. Cette méthodologie est particulièrement appropriée dans le cas d'applications hypermédia hautement structurées et où un haut niveau de volatilité de l'information.

La méthodologie RMM propose une série d'étapes à suivre pour concevoir son application hypermédia, fournit des formalismes supplémentaires de structure d'accès et accroît l'importance de la représentation graphique.

1. Relationship Management Data Model (RMDM)

Un modèle de données est un ensemble d'objets logiques utilisés pour fournir une abstraction d'une partie d'un monde réel. Les modèles de données sont nécessaires pour exprimer la conception d'une application. RMDM fournit un langage pour décrire les objets d'information et les mécanismes de navigation dans les applications hypermédia. La Figure 6.46 illustre les primitives du modèle RMDM.

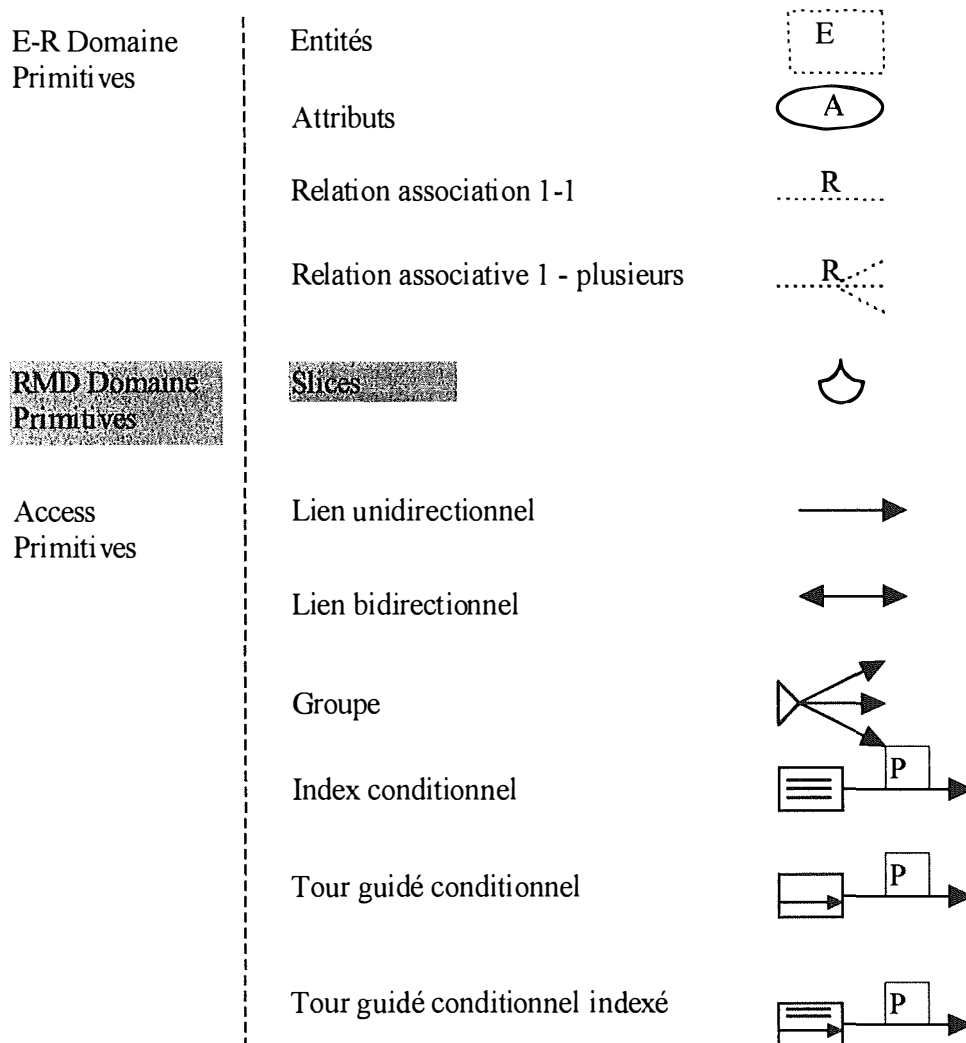


Figure 6.46: Primitives du RMDM

La partie supérieure de la Figure 6.46 illustre les primitives du domaine, lesquelles modélisent l'information à propos du domaine de l'application. Le type entité et leurs attributs représentent des objets abstraits ou physiques. Les relations associatives qui peuvent être 1-1 ou 1-plusieurs décrivent les associations parmi différents types d'entités.

Etant donné que les entités peuvent consister en un grand nombre d'attributs de natures différentes, il pourrait vite devenir impraticable de tous les présenter dans une seule instance d'entité à la fois. Donc les attributs sont regroupés en slices. La représentation graphique se situe dans la partie du milieu de la Figure 6.46.

La navigation est supportée dans RMDM par six primitives d'accès telles que le montre la Figure 6.46. Les liens unidirectionnels et bidirectionnels sont utilisés pour spécifier l'accès entre les slices d'une entité. La navigation à travers différentes entités est permise

grâce aux objets index, tour guidé et groupe. L'index agit comme une table des matières d'une liste d'instances d'entités, fournissant un accès à chacune des instances listées. Un tour guidé construit un chemin linéaire à travers une collection d'items permettant à l'utilisateur de se déplacer aussi bien en avant qu'en arrière sur ce chemin. La construction de groupes est comme un mécanisme de menu qui permet d'accéder à d'autres parties du document hypermédia. Les conditions ou prédiqua logiques qualifiant les index ou les tours guidés déterminent quelles instances d'une entité sont accessibles à partir de la construction.

2. Relationship Management Methodologie (RMM)

La méthode RMM est illustrée graphiquement dans le Figure 6.47 incluant le contexte d'un cycle complet de développement de logiciel.

RMM se focalise sur les phases de conception, développement et construction. Pour des raisons de lisibilité, les flèches de feed-back ne sont pas rajoutées à cette figure, à l'exception des boucles de feed-back pour l'étape de conception RMM proprement dit.

a) 1° Etape : ER Design

La première étape de conception est de représenter le domaine d'information d'une application par un diagramme entité - relation. Cette étape du processus de conception représente une étude des entités nécessaires et des relations dans le domaine d'application. Ces entités et ces relations forment les bases de l'application hypermédia et beaucoup d'entre elles seront représentées dans l'application finale comme des noeuds et des liens dans un hypermédia.

b) 2° Etape : Slice Design

Cette étape est particulière pour les applications hypermédia car elle détermine comment l'information dans une entité sera présentée à l'utilisateur et comment ils peuvent y accéder. Il permet de découper une entité en slices et de les organiser dans un réseau hypertexte.

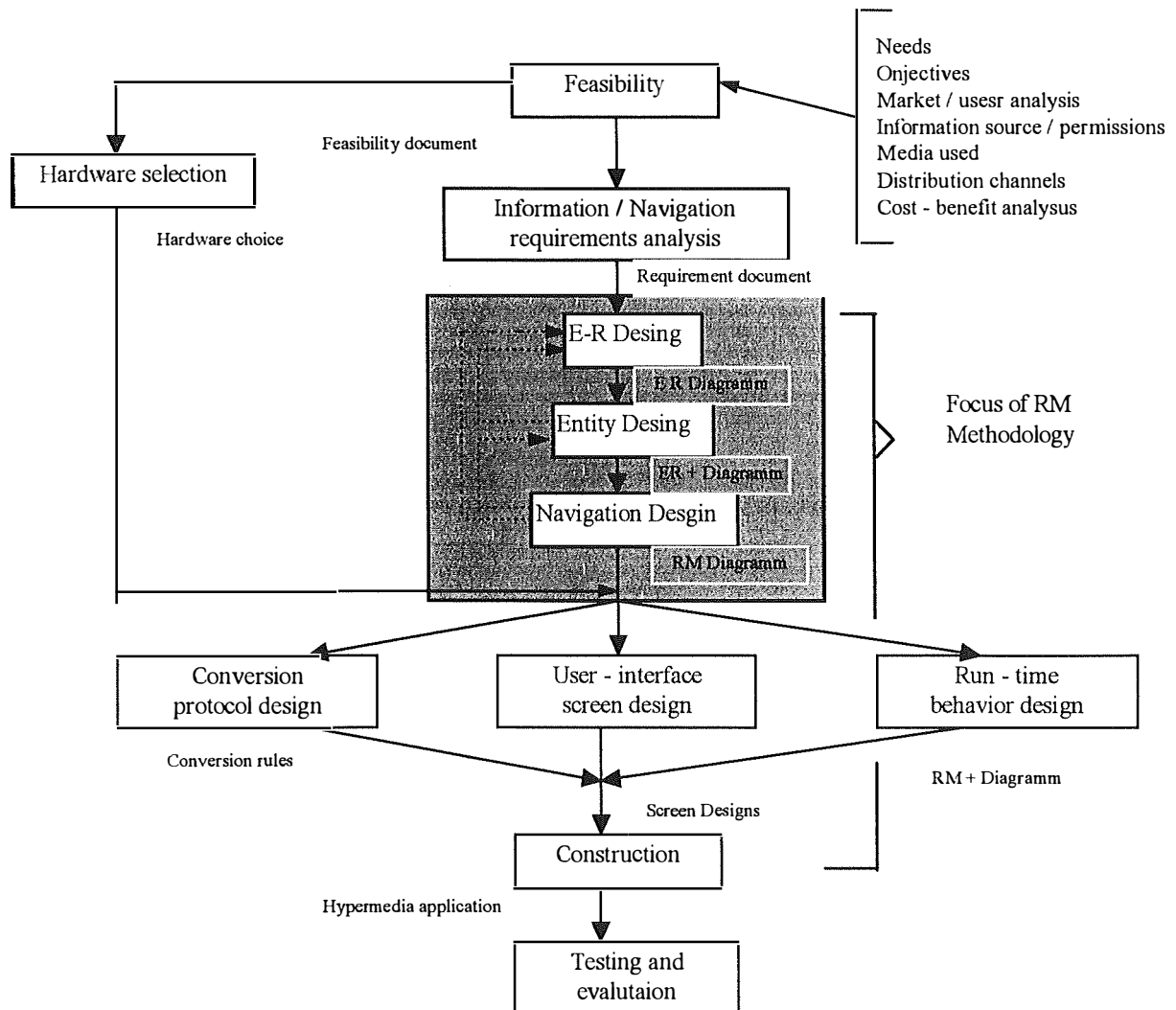


Figure 6.47: la méthode RMM

Bien que dans sa forme la plus simple toute l'information d'une entité peut être représentée dans une seule fenêtre avec des scrollbars, cette approche simpliste pour le développeur est indésirable pour l'utilisateur. Car il peut être désorienté en scrollant de longues fenêtres. C'est pour cela que l'information peut être divisée en unités, qui peuvent être présentées comme séparées mais forment un ensemble inter-relié.

L'organisation d'entités en slices est appelée : *'The Slice Design Phase'*, et son résultat est un diagramme de slices. Chaque slice regroupe un ou plusieurs attributs de l'entité. Chaque entité a un slice particulier, la source ('head'), lequel sera utilisée comme ancre par défaut pour les liens entrant dans cette entité.

Le diagramme des slices modélise aussi la navigation entre les slices utilisant des liens uni et bidirectionnels. Les liens qui représentent les connections entre les slices sont appelés liens structuels. Les liens structuels diffèrent des relations associatives dans le sens où

ces premières connectent des pièces d'information dans la même instance d'une entité alors que les secondes interconnectent différentes instances d'entités.

c) 3° Etape : *Navigational Design*

Dans cette étape, on conçoit les chemins que va permettre la navigation hypertexte. Chaque relation associative apparaissant dans le diagramme illustré à la Figure 6.46 est analysée. Si selon l'analyse des besoins, une relation associative devrait être accessible pour la navigation, elle est remplacée par une ou plusieurs structures d'accès RMDM. Toutes les structures de navigation sont génériques, ce qui permet facilement à l'application d'être mise à jour.

Après, on construit des structures d'accès de haut niveau en groupant des items d'intérêt. L'utilisation de ces structures de groupe fournit un accès hiérarchique par le biais de menu. Durant cette étape, les concepteurs doivent identifier les composants informationnels et les relations qui seront accessibles, les groupes qui seront présentés, et les structures d'accès utilisées dans chaque cas.

d) *Next steps*

L'étape 4, '*Conversion Protocol Design*', utilise un ensemble de règles de conversion pour transformer chaque élément du diagramme RMDM en objet pour la plate-forme cible. Cette étape est actuellement réalisée manuellement par les programmeurs.

L'étape 5, *Users Interface Design*, comprend la conception des écrans pour chaque objet apparaissant dans le diagramme RMDM obtenu à l'étape 3. Ceci inclut l'affichage des boutons, l'apparence des noeuds et index, ainsi que la localisation des aides navigationnelles.

Les décisions à propos de la manière dont les liens sont traversés, l'histoire, le backtracking et les mécanismes navigationnels peuvent être implémentés durant l'étape 6, *Runtime Behavior Design*.

L'étape 7 consiste en la construction et l'étape de test comme dans les projets traditionnels de Software Ingénierie. Dans les applications hypermédia on doit spécialement faire attention et tester tous les chemins navigationnels.

3. Conclusion

La méthodologie de conception et de construction hypermédia (RMM) convient mieux pour les applications qui ont une structure régulière et spécialement quand il y a un besoin fréquent d'update de l'information pour conserver le système courant. Beaucoup d'applications commerciales telles que les catalogues de produits, les applications de commerce électronique, les manuels de conception, etc, rencontrent cette description.

La méthodologie RMM peut servir de base pour la conception et le développement d'applications hypermédia robustes. Le fait que RMM utilise les modèles de conception entités - associations, lui procure un avantage certain. Car la méthode entité-association est une méthode courante, familière, connue et largement documentée. Elle permet de modéliser les dépendances de l'information dans beaucoup de domaines d'application. De plus, dans beaucoup de cas, le modèle entité-association de l'application existe déjà, on peut donc tout simplement la réutiliser pour transformer cette application en hypermédia.

Par contre, la méthodologie RMM procure très peu de flexibilité en ce qui concerne la navigation, et cela aussi bien entre entités qu'entre slices. Il pourrait être avantageux de permettre l'utilisation de structures de navigation à l'intérieur d'une entité, ainsi que de permettre l'utilisation de plus de structures de navigation telle que celle définie dans le chapitre 4. Cette méthodologie ne permet pas de concevoir des applications hypermédia interactive. Elle ne conçoit pas l'échange d'information, l'utilisation de formulaire, ... Bref, seule des applications statiques, non interactives peuvent être conçues. Aucune prise en compte de règles ergonomiques n'est intégrée dans cette méthode alors que pour la conception de l'interface homme-machine des applications, l'ergonomie est certainement le plus important. Enfin, il serait intéressant de donner une brève description de la composition des slices dans RMM. Ne serait-ce que pour déterminer les bases des étapes 5 et 6.

B. Méthodologie TRIDENT

Cette méthodologie a été développée par l'équipe TRIDENT [Trident, 1994a], [Trident, 1994b] menée par le Professeur François Bodart. Cette méthodologie est dédiée au développement d'applications de gestion hautement interactives, centrées sur la tâche de l'utilisateur.

Cette méthodologie de développement présente les fondements d'une méthode reposant sur une analyse en profondeur du processus pour garantir la production d'applications interactives de qualité, tant au plan ergonomique qu'au plan technique. Pour ce faire, on propose cinq dimensions clés :

1. la formalisation des spécifications IHM à partir de l'analyse de la tâche
2. le pilotage actif de la présentation à partir de règles ergonomiques
3. la dérivation de l'architecture de l'application interactive à partir de l'analyse de la tâche et des composants de la présentation
4. la spécification d'un modèle de la conversation de haut niveau à partir de l'analyse de la tâche.
5. la particularisation du cadre de travail méthodologique en un cadre de travail de spécification

1. La formalisation des spécifications IHM à partir de l'analyse de la tâche

Partant d'une analyse complète de la tâche, en ce compris la caractérisation des stéréotypes d'utilisateur et de l'environnement de travail, l'objectif est de proposer une démarche systématique de spécification des composants IHM à partir du produit de cette analyse de la tâche.

a) Analyse de la tâche

Cette analyse s'effectue selon la méthode TKS (Task Knowledge Structur) provenant de la psychologie cognitive. Cette méthode est moins précise que la CCT qui spécifie toutes les actions mentales et sensori-motrices mais est moins rudimentaire que la simple découpe hiérarchique de HTA.

On décompose donc la tâche qui traduit le but principal à atteindre. Ce but est décomposé en sous-buts à atteindre. Chaque sous-but peut être réalisé par des procédures composées d'actions portant sur des objets. La manière appropriée pour atteindre un but, résulte d'une activité de planification ou de résolution de problèmes.

Les objets sur lesquels portent les actions élémentaires peuvent être informationnel, conceptuel ou physique.

b) Expression du produit de l'analyse de la tâche

(1) Rédaction d'un schéma entité-association orienté-objet

Les objets produits par l'analyse de la tâche et les relations entre ceux-ci sont spécifiés dans un schéma entité-association orienté-objet. Ce schéma forme le pont entre le monde cognitif de l'analyse de la tâche et le monde conceptuel de l'ingénieur système.

(2) Identification des fonctions sémantiques de l'application

Ce processus qui a pour but d'obtenir les spécification fonctionnelles de l'applications, transforme les actions en fonctions par une sorte d'abstraction informatique. Ces fonctions deviendront les méthodes attachées aux objets de l'application.

(3) Rédaction d'un graphe d'enchaînement des fonctions

Après avoir décrit les aspects statiques de la tâche, il reste à en décrire les aspects dynamiques. Le graphe d'enchaînement des fonctions représente le modèle de comportement de la tâche. Ce graphe (dont les primitives sont représentées à la Figure 48) montre l'enchaînement fonctionnel et l'échange d'informations entre les fonctions mais ne préjuge en rien d'un ordre d'exécution.

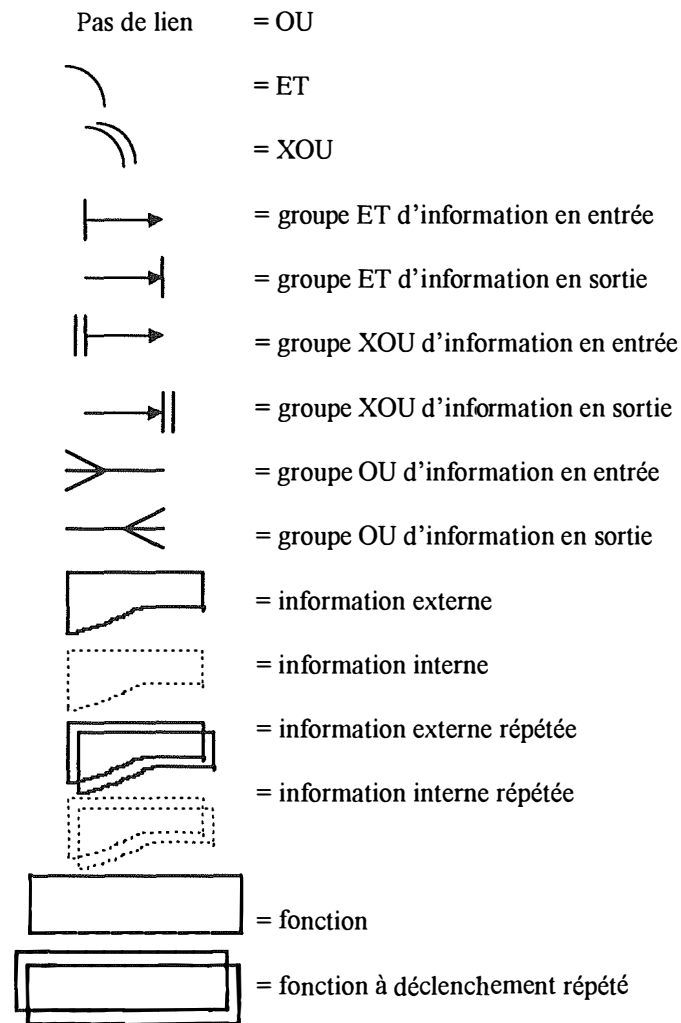


Figure 48: Primitives du graphe d'enchaînement

Chaque fonction reçoit des informations en entrée, représentant les données nécessaires à la bonne exécution de la fonction. Chaque fonction peut produire des informations en sortie qui sont soit externes (destinées à l'utilisateur), ou soit interne (destinée à d'autres fonctions).

(4) Choix des attributs de dialogue

On choisit le ou les styles d'interaction considérés comme appropriés à cette tâche interactive. Ce choix est orienté par les paramètres relatifs à la tâche interactive, aux stéréotypes d'utilisateur, à l'environnement de travail et les styles d'interaction préconisés.

A partir de ce style d'interaction, on dérive quatre attributs de dialogue : le mode de dialogue (séquentiel, asynchrone et mixte), le contrôle de dialogue (interne, externe,

mixte), le type de déclenchement des fonctions (automatique, manuel implicite ou explicite, affiché ou non) et la métaphore (basée sur la conversation, le mini-monde ou les deux).

2. Le pilotage actif de la présentation à partir de règles ergonomiques

Cette dimension a pour objectif de fonder une démarche systématique de spécification des composants de présentation de l'interface homme-machine sur un ensemble prédéterminé de règles ergonomiques en vue de satisfaire un certain nombre de critères ergonomiques pertinents pour la tâche.

a) Identification des Unités de Présentation (UP)

A chaque sous-tâche de la tâche interactive, correspond une UP. Cette correspondance s'effectue donc au moyen de critères d'identification de sous-tâche dans l'analyse de la tâche. Une UP peut donc se matérialiser comme un sous-graphe du graphe d'enchaînement étant donné que ce dernier représente l'enchaînement des fonctions au sein d'une tâche.

b) Identification des fenêtres

On décompose chaque UP en fenêtre. Les fenêtres ne peuvent être identifiées correctement au moyen de règles ergonomiques précises. Cependant il existe certains critères d'identification des fenêtres tels que l'identification minimale, maximale, entrée-sortie, fonctionnelle, libre ou encore groupée. [Trident, 1994b]

c) Sélection des Objets Interactifs Abstraits (OIA)

A chaque fenêtre logique, on fait correspondre une fenêtre physique, une boîte de dialogue ou un panneau selon le style d'interaction retenu. Au sein de chaque fenêtre, on fait correspondre un objet interactif abstrait simple, ou composite, à chaque information en entrée ou en sortie et à chaque fonction. Les règles de correspondance sont issues du '*corpus minimal des applications de gestion*' [Vanderdonckt, 1992b].

d) Transformation des OIA en Objets Interactifs Concrets (OIC)

A chaque OIA, on fait correspondre l'objet interactif concret correspondant à l'environnement physique.

e) Placement des OIC

On définit rigoureusement le placement les coordonnées de la position de chaque OIC au sein d'une fenêtre en s'appuyant sur des règles ergonomiques de placement [Vanderdonckt, 1994].

f) Edition de la présentation

On édite, corrige et améliore manuellement le résultat de la génération automatique en adaptant la présentation aux besoins des utilisateurs.

3. La dérivation de l'architecture de l'application interactive à partir de l'analyse de la tâche et des composants de la présentation

Cette dimension a pour but de construire l'architecture de l'application tout en respectant l'analyse de la tâche.

a) Contenu du modèle

Ce modèle d'architecture comporte trois classes d'objets :

1. la classe des objets de contrôle (OC) est une classe qui se décompose en OC de plusieurs types qui gèrent le dialogue et assurent la correspondance entre les structures de données de l'application et celles de la présentation. Chaque OC possède un comportement propre qui prend la forme d'un script.
2. la classe des objets de l'application (OA) n'est pas décomposable, ses éléments représentent les fonctions de l'application.
3. la classe des objets interactifs (OI) est une classe générique qui se décompose en OI de deux types, les OIC issus de l'application traduisant les informations en entrée-sortie des fonctions et les OIC induits strictement par le dialogue.

b) Construction de la hiérarchies des OC

A chaque étape de la composition de la présentation, on fait correspondre un OC. Voici les objets de contrôle proposés :

OC-TI : OC correspondant à la Tâche Interactive

OC-UP : OC correspondant aux UP

OC-F : OC correspondant aux Fenêtres racines

OC-Fc : OC correspondant aux fonctions de l'application (toujours fils de OC-F)

c) Construction de la hiérarchie des OI

Les objets interactifs peuvent être composites ou simples. Les OI composites correspondent aux fenêtres ainsi qu'au regroupement d'OI simples. qui quant à eux, représentent les données de saisie et d'affichage (il s'agit des OI d'entrée sortie OI E/S) ainsi que les éléments de contrôle (OI induits par la présentation (OI-P)).

Ces objets correspondent à des OIC proposés par les environnements physiques. Les OI sont toujours fils d'un OC-F à qui ils renvoient les événements significatifs de son état ainsi que lui renvoyer les valeurs dont il serait porteur.

4. La spécification d'un modèle de la conversation de haut niveau à partir de l'analyse de la tâche.

La méthode proposée consiste à définir la conversation par raffinements successifs en partant des niveaux les plus élevés pour finir avec les niveaux les plus bas. Les conversations peuvent être définies à l'aide d'un script.

a) Contenu de la conversation

La conversation de l'IHM s'explique à trois niveaux :

1. le niveau inter UP : comment déclencher les différentes sous-tâches matérialisées par les UP au niveau d'une tâche interactive ? Elle peut se formuler à l'aide d'un schéma. Elle peut aussi se formuler à l'aide d'un graphe qui précise les contextes de déclenchement, de déroulement et de terminaison des UP.
2. le niveau intra-UP : comment enchaîner les différentes fenêtres entre elles au niveau d'une unité de présentation ? Elle peut être formulée à l'aide d'un graphe d'enchaînement des fenêtres exprimé en terme d'opérations propres au multi-fenêtrage. En plus de l'enchaînement, on peut aussi décider de la configuration et de leur placement à l'écran.

3. le niveau intra-fenêtre : quelles sont les dépendances de comportement des OIC au sein d'une même fenêtre ? La conversation inter-fenêtre peut être formulée par un graphe d'interaction des OIC d'une fenêtre à l'aide des différentes primitives offertes par les OIC. Cette conversation doit s'inspirer des quatre attributs de dialogue cités pour chaque UP.

5. La particularisation du cadre de travail méthodologique en un cadre de travail de spécification

Cette cinquième dimension de la méthodologie ne sera ici que très peu détaillée. En effet, cette cinquième dimension est présente pour particulariser le cadre méthodologique à un cadre de travail particulier, spécifique. Or ce cadre de travail spécifique dépend de la classe de problèmes traités, du style d'interaction et de certaines hypothèses propres. Ces spécificités ont des impacts sur la spécification et la génération de la présentation, sur l'architecture, sur la conversation, ...

6. Conclusion

Cette méthodologie est orientée vers des applications de gestion hautement interactive. Elle offre les avantages d'obtenir une interface mieux orientée vers la tâche finale à réaliser. Elle permet de prendre en compte toutes les facettes de la création d'une application interactive, partant de l'analyse jusqu'à la conception des écrans, des objets à utiliser, de leur placement et surtout du comportement de tous les objets entre eux, de leur conversation ... Un avantage non négligeable de cette méthodologie est son respect à tous niveaux, de règles ergonomiques, que ce soit pour le choix des objets à utiliser que pour le design de l'écran [Vanderdonckt, 1994].

Par contre, dans le cadre d'application hypermédia, cette méthodologie ne permet pas de s'occuper correctement de la navigation. La navigation découle de l'enchaînement des écrans, et donc, de manière implicite, de l'analyse de la tâche et de sa découpe en UP et fenêtres. Cette méthode ne permet donc pas de facilement créer un index, de le modifier, ...

C. Perspectives d'une méthodologie dédiée à la conception d'applications hypermédia

Les deux méthodologies présentées possèdent chacune des avantages dont il faudrait tenir compte pour créer une nouvelle méthodologie. De plus, on pourrait voir une

certaine complétude entre elles. En effet, la méthode RMM est d'ores et déjà orientée hypermédia grâce à son modèle RMDM mais ne propose que des pistes pour la création de la présentation, le comportement à l'exécution, ... ce qui est justement la force de la méthodologie TRIDENT.

Dès lors, nous proposons de suivre le schéma de la Figure 6.47 pour construire cette méthodologie. La première phase serait la construction du schéma entité-association pour représenter les domaines d'informations de l'application. En ce qui concerne la seconde phase, considérons que chaque entité informationnelle représente une unité de présentation au sens où TRIDENT l'entend. La seconde phase consisterait donc en la phase d'identification des fenêtres pour représenter les attributs des entités. Cette phase remplacerait la phase de conception des slices de RMM que nous avons jugée trop légère dans sa méthodologie et sa rigueur. Ensuite viendrait celle de la conception de la navigation comme le propose RMM. Finalement, on transformerait les éléments du modèle RMDM en objets pour l'environnement de destination tous comme le préconise la phase 4 de RMM.

Par contre les phases suivantes seraient inspirées de celles de TRIDENT. Il s'agirait de la phase de conception des écrans à l'aide d'objets. Mais ces objets seraient principalement des objets informationnels (au sens où nous les avons définis dans le chapitre 4) ainsi que des objets interactifs. Le choix des objets à sélectionner proviendra aussi du suivi de règles ergonomiques telles que celles présentées dans [Vanderdonckt, 1994]]. Ensuite, on définirait la conversation et le comportement des fenêtres et des objets qui les composent.

Enfin, la construction se réaliserait tout comme pour les projets d'application traditionnelle. L'utilisation de l'environnement de création définit dans ce mémoire faciliterait la construction de telles applications. De plus l'existence de structures de navigation facilite grandement la transformation des primitives de navigation utilisées par RMDM.

II. Conclusion

Ce travail qui a conduit au développement d'un environnement de création d'application hypermédia pour le projet SYRECOS a permis d'aborder des domaines fort diversifiés.

Pour mener à bien ce projet, nous avons présenté globalement les systèmes hypermédia, les grandes applications de ces systèmes. Vu que le projet SYRECOS a pour objectif de créer un INTRANET, nous avons étudié les technologies WEB actuelles qui peuvent nous aider à la réalisation de ce projet.

Dans le cadre de la réalisation de l'environnement de développement d'applications hypermédia, nous avons défini les concepts composants notre système hypermédia et le rôle de chacun des acteurs qui seront appelés à participer dans la réalisation d'une application hypermédia dans le cadre du projet SYRECOS.

Dans le chapitre 5, nous avons réalisé les bases de l'environnement de développement. Il nous est apparu inutile de recréer tous les outils nécessaires à l'environnement, c'est pour cette raison que nous nous basons sur un outil développé par Microsoft : FRONTPAGE. En plus de cet outil, nous avons réalisé des outils spécifiques qui faciliteront la création de scénarios hypermédia.

Finalement, nous avons tenté de définir les grandes lignes d'une méthodologie de développement d'applications hypermédia. Cette méthodologie se base sur la fusion de la méthode RMM et TRIDENT.

Nous pensons, de cette manière, avoir tracé les grandes lignes des différents aspects que touchent la création d'applications hypermédia. Ce mémoire ne se veut pas avoir la prétention d'avoir bouclé le sujet, que du contraire, il a juste voulu poser les premières briques d'un projet ambitieux et prometteur pour l'avenir, tout en ouvrant la voie à plusieurs autres mémoires qui pourraient se spécialiser dans l'élaboration de l'environnement ou de la méthodologie.

BIBLIOGRAPHIE

- [Acksyn, 1988] Robert M. Acksyn, Douglas L. McCracken, Elise Yoder, *KMS : A Distributed Hypermedia System for Managing Knowledge in Organizations*, Communications of the ACM, July 1988
- [Baird and Percival, 1989] P. Baird and M. Percival *Glasgow Online : Database development using Apple's Hypercard*. Hypertext : Theory into Practice, Ablex, 75-92
- [Berners-Lee, 1994] T. Berners-Lee, L. Masinter, M. McCahill : *Uniform Resource Locators (URL)*, Internet Network Working Group RFC 1738, CERN, Xerox PARC, University of Minnesota, December 1994
- [Berners-Lee, 1995a] T. Berners-Lee, D. W. Conolly, *Hypertext Markup Language - 2.0*, Internet Network Working Group RFC 1866, MIT/W3C, November 1995
- [Berners-Lee, 1995b] T. Berners-Lee, R. T. Fielding, H. F. Nielsen : *HyperText Transfer Protocol - HTTP1.0 IETF draft* - Work in progress (expires April 14, 1996), MIT/LCS, UC Irvine, CERN, October 14, 1995
- [Bigelow, 1988] J. Bigelow *Hypertext and CASE*. IEEE Software 5,2 March 1988
- [Brown, 1987] Peter J. Brown, *Turning Ideas into Products : The Guide System*, Hypertext '87 Proceedings, November 1987.
- [Bush, 1945] Vannevar Bush, *As We May Think*, The Atlantic Monthly, July 1945.
- [Campagnoni and Ehrlich, 1989] F.R. Campagnoni and K. Ehrlich *Information retrieval using a hypertext-based help system*, ACM SIGIR'89, June 1989
- [CGI 1.1] The common Gateway interface, NCSA 1995, URL :
<<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>>

- [Conklin and Begeman, 1988] J. Conklin and M.L. Begeman *A hypertext tool for exploratory policy discussion* Trans. Office information systems, October 1988
- [Crane, 1987] G. Crane *From the old to the new : Integrating hypertext into traditional scholarship* Hypertext '87 Proceedings, November 1987
- [Crane, 1988] G. Crane *Redefining the book : Some preliminary problems* Academic Computing, February 1988
- [De Young, 1989] L. De Young *Hypertext challenges in the auditing domain.* Hypertext '89, November 1989
- [Engelbart, 1963] Douglas C. Engelbart, *A Conceptual Framework for the Augmentation of Man's Intellect*, Vistas In Information Handling, Volume 1, Spartan Books, Washington D.C., 1963.
- [France, 1994] M. France *Smart contracts* Forbes ASAP, August 1994
- [Halasz., 1987] F.G. Halasz, T.P.Moran and R.H.Trigg *NoteCards in a nutshell* CHI+GI '87, April 1987
- [Halasz, 1988] Franck G. Halasz, *Reflections on NoteCards : Seven issues for the next generation of hypermedia system.* Communications of the ACM 31, 7 (JULY), pp 836-852.
- [Halasz, 1988b] Franck G. Halasz, *Notecards, A Multimedia Idea Processing Environnement*, Interactive Multimedia edited by Ambron, Sueann and Hooper, Kristina, Microsoft Press, 1988.
- [Hodges, 1989] M.E. Hodges, R.M.Sasnett and M.S. Ackerman *A construction set for multimedia applications* IEEE Software 6,1 January 1989
- [HTML RM, 1996] HTML Reference Manual,, 1996, URL :
[http ://www.sandia.gov/sci_compute/html_ref.html](http://www.sandia.gov/sci_compute/html_ref.html)

- [Isakowitz, 1995], T. Isakowitz, E.A. Stohr, P. Balasubramanian :*RMM : A Methodology for Structured Hypermédia Design*, Communications of the ACM, August 1995, Vol. 38, N°8, pp34-44
- [Krol and Hoffman, 1993] E. Krol, E. Hoffman, *FYI on « What is the Internet ? »*, Internet Network Working Group, RFC 1462, Mai 1993
- [Leggett., 1989] J. Legett, J.L.Schnase and C.J. Kacmar *A short course on hypertext. Technical report TAMU 89-004 Texas A&M University, January 1989*
- [Marchionini and Crane, 1994] G. Marchionini and G. Crane *Evaluating hypermedia and learning : Methods and results from the Perseus Project*. Trans. Information system, January 1994
- [Meyrowitz, 1986] Norman K. Meyrowitz, *Intermedia : The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework*, OOPSLA '86 Proceedings, 1986.
- [Mylonas, 1992] E. Mylonas An interface to classical Greek civilization J. Of the American Society for Information Science 43,2 March 1992
- [Nelson, 1980] Ted Nelson, *Replacing the Printed Word, A Complete Literacy System*, Information Processing '80, 1980.
- [Nelson, 1987] Ted Nelson, *All for One and One for All*, Hypertext'87 Proceedings, November 1987.
- [Nielsen,1995] Jakob Nielsen. *Multimedia and Hypertext : The Internet and beyond*, AP Professional, 1995.
- [Pejtersen, 1989] A.M.Pejtersen *A library system for information retrieval based on a cognitive task analysis and supported by an icon-based interface*. SIGIR '89, June 1989
- [Perkins, 1995] R.Perkins :*The interchange online network : Simplifying information access* CHI '95

- [Schnase and Leggett, 1989] J.L.Schnase and J.J.Leggett *Computational hypertext in biological modeling* Hypertext '89, November 1989
- [Schneiderman, 1988] Ben Schneidermann, *User Interface Design for the HyperTies Electronic Encyclopedia*, Hypertext '87 Proceedings, November 1987.
- [Smith and Weiss, 1988] John Smith, Stephen Weiss. *An Overview oh Hypertext*, Communication of the ACM, July 1988.
- [Streitz, 19992] N.Streitz, J.Haake, J.Hannemann, A.Lemke, W.Schuler, H.Schütt and M.Thüring *SEPIA : A cooperative hypermedia authoring environnement* ECHT '92, Decmeber 1992
- [Trident, 1994a] F..Bodart, A-M. Hennebert, J-M. Leheureux, I. Provot, J. Vanderdonckt, G Zucchinietti :*Dimensions clé pour une méthodologie de développement d'applications interactives*. International Eurographics Workshop on Design, Specificaton and Verification of Interactive Systems, Bocca di Magra (La Spezia), 8-10 June 1994
- [Trident, 1994b] F..Bodart, A-M. Hennebert, J-M. Leheureux, I. Provot, J. Vanderdonckt, G Zucchinietti, F. Taes : *Tâche interactive de l'enregistrement d'un bon de commande client : étude de cas*, Institut d'informatique, Projet Trident, 30 novembre 1994
- [Vanderdonckt, 1992a] J. Vanderdonckt :*Une description Orientée objet des objets interactifs abstraits utilisés en interface Homme-Machine*, Institut d'Informatique, 1991
- [Vanderdonckt, 1992b] J. Vanderdonckt :*Corpus ergonomique minimal des applications de gestion*, Institut d'Informatique, octobre 1992
- [Vanderdonckt, 1994] J. Vanderdonckt :*Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur, Namur, 1994

[Wilson, 1988] K.S.Wilson *Palenque : An interactive multimedia digital video interactive prototype for children* ACM CHI '88, May 1988

[Yankelovich, 1988] Nicole Yankelovich, Bernard J. Haan, Norman K. Meyrovich, Steven M. Drucker, *Intermedia : The Concept and the Construction of a Seamless Information Environnement*, IEEE Computer, January 1988.

ANNEXES 1

GRAMMAIRE BNF DU LANGAGE DE DESCRIPTION

```
<vide> ::=
<lettre> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|
  B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<chiffre> ::= 0|1|2|3|4|5|6|7|8|9
<caractère special> ::= _|+|-|*|/|( | )|=|,|.|»|!|'|?|<|>|:
<caractère> ::= <lettre>|<chiffre>|<caractère special>
<identifiant> ::= <caractère> [<identifiant>|<vide>]
<phrase> ::= <identifiant> [ <identifiant>| | <vide>]
<nombre> ::= <chiffre> [<chiffre> | <vide>]
<scenario> ::= scenario <identifiant> ;
  <description>
  titre <phrase> ;
  <bloc>
<description> ::= description <phrase> ;
<bloc> ::= <ensemble page> ; <navigation>
<ensemble page> ::= <page> ; [<ensemble page>|<vide>]
<page> ::= page <identifiant> ; <description> <titre>
  <corps page>
<titre> ::= titre <phrase> ;
<Frameset> ::= Frameset <identifiant> ;
  colonne <liste pourcentage> ;
  <liste rangées> ;
<liste rangées> ::= rangée <liste pourcentage> ;
  <liste frame> ;
<liste pourcentage> ::= <numero>% [, <liste pourcentage>|<vide>]
<liste frame> ::= frame <page> [ ;<liste frame>|<vide>]
<corps page> ::= [<section>|<formulaire>] ;
  [<corps page> | <vide>]
<section> ::= section <identifiant> ; <description>
  <liste d'objets informationnel>
```

```

<liste d'objets informationnel> ::= <objet informationnel> ;
    <liste objet informationnel> | <vide>
<objet informationnel> ::= <alignement> <ordre> <type objet info>
<alignement> ::= alignement <choix alignement> ;
<choix alignement> ::= gauche | centre | droite
<ordre> ::= ordre <nombre> ;
<type objet info> ::= <texte> | <liste> | <tableau> | <video>
    | <son> | <icone> | <image>
<texte> ::= texte <identifiant> ;
    [alignement <choix alignement texte> ; | <vide>]
    [couleur <identifiant> ; | <vide>]
    [police <identifiant> ; | <vide>]
    [style <identifiant> ; | <vide>]
    [taille <identifiant> ; | <vide>]
    <liste partie>
<choix alignement texte> ::= gauche | droite | centre | justifie
<liste partie> ::= <partie> ; [<liste partie> | <vide>]
<partie> ::= partie <identifiant> ;
    [<description>|<contenu partie>]
    [<style partie>|<vide>]
    [destination [<identifiant>|<vide>] | <vide>]
<contenu partie> ::= contenu <phrase> ;
<style partie> ::= [couleur <identifiant> ; | <vide>]
    [police <identifiant> ; | <vide>]
    [style <identifiant> ; | <vide>]
    [taille <identifiant> ; | <vide>]
<liste> ::= liste <identifiant> ; <description>
    [type <choix type liste> ; | <vide>]
    <liste type objet info>
<choix type liste> ::= puce | numero
<liste type objet info> ::= [<type objet info> <liste type objet
    info>] | <vide> ;
<tableau> ::= tableau <identifiant> ; <description>
    [<vide> | nbr colonne <nombre> ;]
    [<vide> | nbr ligne <nombre> ;]
    <liste cellule>]
<liste cellule> ::= [<cellule> ; <liste cellule>] | <vide>
<cellule> ::= numero ligne <nombre> ;
    numero colonne <nombre> ;
    <type objet info>
<icone> ::= icone <identifiant> ; <description>
    [<taille> | <vide>]

```

```

    [destination [<identifiant>|<vide>] | <vide>]
<taille> ::= hauteur <nombre> ; largeur <nombre> ;
<image> ::= image <identifiant> ; <description>
    [<taille> ; | <vide>]
    [icone <identifiant> ; | <vide>]
    [destination [<identifiant>|<vide>] | <vide>]
<son> ::= son <identifiant> ; <description>
    [icone <identifiant> ; | <vide>]
<video> ::= vide <identifiant> ; <description>
    [icone <identifiant> ; | <vide>]
<formulaire> ::= formulaire <identifiant> ;
    <liste objet>
    <submit> <reset>
<submit> ::= submit ; methode <methode> ; action <phrase>
    [libelle <phrase> ; | <vide>]
<methode> ::= get | post
<reset> ::= reset ;
    [libelle <phrase> ; | <vide>]
<liste objet> ::= <objet>      [<liste objet> | <vide>]
<objet> ::= <objet informationnel> | <objet interactif>
<objet interactif> ::= <alignement> <ordre> <tab_orders>
    [<editbox> | <multibox> | <listbox> |
    <combobox> | <togglebutton>]
<tab_orders> ::= <nombre> ;
<editbox> ::= editbox <identifiant> ; <description>
    [libelle <phrase> ; | <vide>]
    long-max <nombre> ;
    long-champ <nombre> ;
    type-donnee <type donnee> ;
    [valeur <phrase> ; | <vide>]
<type donnee> ::= texte | password
<multibox> ::= multibox <identifiant> ; <description>
    [libelle <phrase> ; <vide>]
    long-max <nombre> ;
    nbr-ligne <nombre> ;
    nbr-colonne <nombre> ;
    [valeur <phrase> ; | <vide>]
<listbox> ::= listbox <identifiant> ; <description>
    [libelle <phrase> ; | <vide>]
    nbr-visible <nombre> ;
    multiple <booleen> ;
    [<liste item> | <vide>]

```

```

<booleen> ::= oui | non
<liste item> ::= <item> ; [<liste item> | <vide>]
<item> ::= item <phrase> ;
           checked <booleen> ;
<combobox> ::= combobox <identifiant> ; <description>
           [libelle <phrase> ; | <vide>]
           [<liste item> ; | <vide>]
<togglebutton> ::= togglebutton <identifiant> ; <description>
           [libelle <phrase> ; | <vide>]
           type [radio | check] ;
           alignement [horizontal | vertical] ;
           [<liste item> ; | <vide>]
<navigation> ::= navigation <ensemble lien>
<ensemble lien> ::= <lien> ; [<ensemble lien> | <vide>]
<lien> ::= lien <identifiant> ;
           source <identifiant> ;
           destination <destination> ;
           position <position lien> ;
           type <type lien> ;
           [libelle <phrase> | <description>]
<position lien> ::= debut | fin | debut-fin
<type lien> ::= texte | bouton
<destination> ::= [page | section] <identifiant>

```

ANNEXE 2

GRAMMAIRE BNF DE L'OUTIL DE TYPAGE

```
<page> ::= page <identifiant> ; <description> <titre>  
          [<type page>|<corps page>]  
<type page> ::= type <identifiant> ;
```

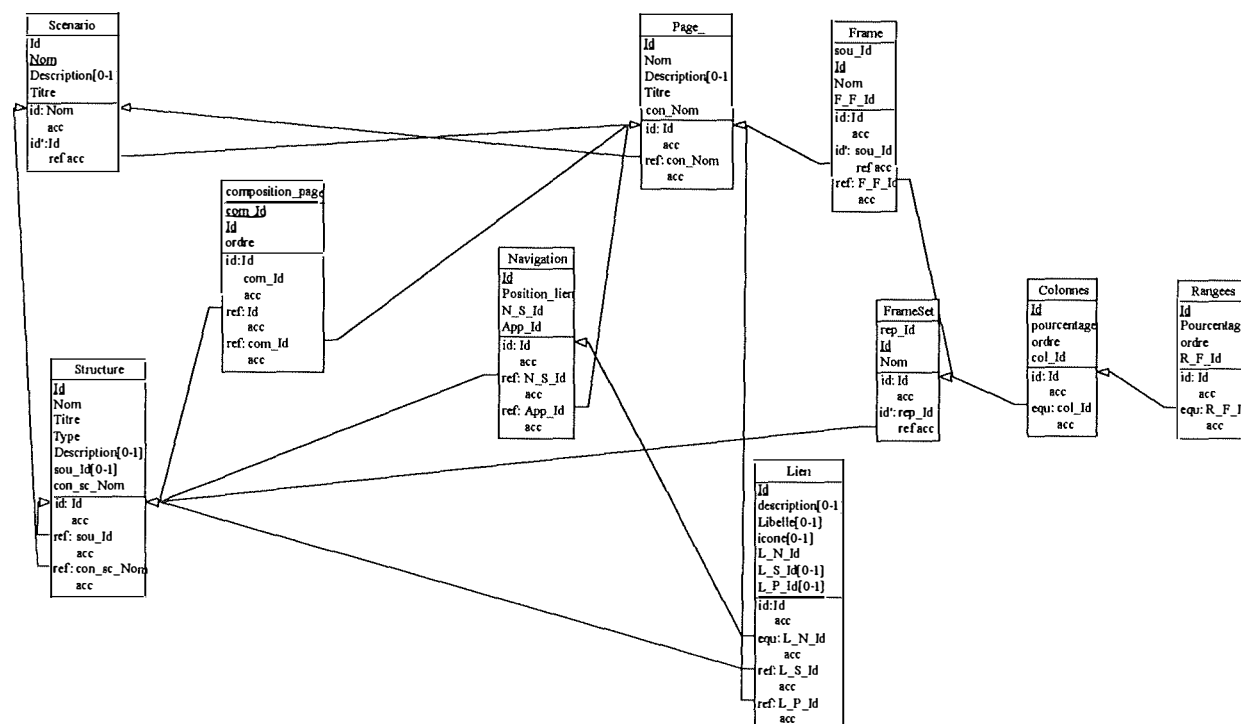
ANNEXE 3

GRAMMAIRE BNF DE L'OUTIL DE NAVIGATION

```
<navigation> ::= <type navigation> [<navigation> | <vide>]
<type navigation> ::= <lien> | <structure>
<lien> ::= lien <identifiant> ;
           source <identifiant> ;
           destination <identifiant> ;
           position <position lien> ;
           type <type lien> ;
           [libelle <phrase> | <description>]
<position lien> ::= debut | fin | debut-fin | gauche | droite |
                  haut | bas
<type lien> ::= texte | icone
<structure> ::= <type structure> <identifiant> ;
               [representation frame ; | <vide>]
               navigation <type navigation> ;
               position <position lien> ;
               type <type lien> ;
               <liste destination>
<type structure> ::= index | guide | sequence | mixte | global
<type navigation> ::= uni | bi
<liste destination> ::= <destination> [<liste destination> |
                                     <vide>]
<destination> ::= [page <identifiant> | structure <identifiant>];
                 [libelle <phrase> ; | <description>]
```

ANNEXE 4

SCHÉMA PHYSIQUE DE LA DB POUR LA NAVIGATION



ANNEXE 5

GÉNÉRATION SQL DE LA DB POUR LA NAVIGATION

```
-- *****
-- * Standard SQL generation      *
-- *****
```

```
-- Database Section
```

```
-- _____
```

```
create database navigation;
```

```
-- DBSpace Section
```

```
-- _____
```

```
-- Table Section
```

```
-- _____
```

```
create table Colonnes (
  Id numeric(1) not null ,
  pourcentage numeric(1) not null ,
  ordre numeric(1) not null ,
  col_Id numeric(1) not null ,
  primary key (Id));
```

```
create table composition_page (
  com_Id numeric(1) not null ,
  Id numeric(1) not null ,
  ordre numeric(2) not null ,
  primary key (Id, com_Id));
```

```
create table Frame (
  sou_Id numeric(1) not null ,
  Id numeric(1) not null ,
  Nom char(30) not null ,
  F_F_Id numeric(1) not null ,
  primary key (Id) ,
  unique (sou_Id));
```



```
create table FrameSet (
    rep_Id numeric(1) not null ,
    Id numeric(1) not null ,
    Nom char(30) not null ,
    primary key (Id) ,
    unique (rep_Id));
```

```
create table Lien (
    Id numeric(1) not null ,
    description long varchar ,
    Libelle char(255) ,
    icone long varchar ,
    L_N_Id numeric(1) not null ,
    L_S_Id numeric(1) ,
    L_P_Id numeric(1) ,
    primary key (Id));
```

```
create table Navigation (
    Id numeric(1) not null ,
    Position_lien char(8) not null ,
    N_S_Id numeric(1) not null ,
    App_Id numeric(1) not null ,
    primary key (Id));
```

```
create table Page_ (
    Id numeric(1) not null ,
    Nom char(30) not null ,
    Description long varchar ,
    Titre char(80) not null ,
    con_Nom char(30) not null ,
    primary key (Id));
```

```
create table Rangees (
    Id numeric(1) not null ,
    Pourcentage numeric(1) not null ,
    ordre numeric(1) not null ,
    R_F_Id numeric(1) not null ,
    primary key (Id));
```

```
create table Scenario (
    Id numeric(1) not null ,
    Nom char(30) not null ,
    Description long varchar ,
    Titre char(80) not null ,
    primary key (Nom) ,
    unique (Id));
```

```
create table Structure (
    Id numeric(1) not null ,
    Nom char(30) not null ,
```

```

Titre char(80) not null ,
Type char(8) not null ,
Description long varchar ,
sou_Id numeric(1) ,
con_sc_Nom char(30) not null ,
primary key (Id));

-- Constraints Section
-- _____

--alter table Colonne add constraint
--  check(exists(select * from Rangees
--              where Rangees.R_F_Id = Id));

alter table Colonne add constraint FKcolonne
  foreign key (col_Id)
  references FrameSet;

alter table composition_page add constraint FKest_compose_de
  foreign key (Id)
  references Structure;

alter table composition_page add constraint FKcompose
  foreign key (com_Id)
  references Page_;

alter table Frame add constraint FKsource_Frame
  foreign key (sou_Id)
  references Page_;

alter table Frame add constraint FKdecomposition
  foreign key (F_F_Id)
  references FrameSet;

--alter table FrameSet add constraint
--  check(exists(select * from Colonne
--              where Colonne.col_Id = Id));

alter table FrameSet add constraint FKrepresentation_frameset
  foreign key (rep_Id)
  references Structure;

alter table Lien add constraint FKsource
  foreign key (L_N_Id)
  references Navigation;

alter table Lien add constraint FKDestination_structure
  foreign key (L_S_Id)
  references Structure;

```

```

alter table Lien add constraint FKDestination_Page
    foreign key (L_P_Id)
    references Page_;

--alter table Navigation add constraint
--    check(exists(select * from Lien
--                where Lien.L_N_Id = Id));

alter table Navigation add constraint FKRepresentation_page
    foreign key (N_S_Id)
    references Structure;

alter table Navigation add constraint FKAppartenance
    foreign key (App_Id)
    references Page_;

alter table Page_ add constraint FKconstit_s_P
    foreign key (con_Nom)
    references Scenario;

alter table Rangees add constraint FKRangee
    foreign key (R_F_Id)
    references Colonnes;

alter table Scenario add constraint FKDepart
    foreign key (Id)
    references Page_;

alter table Structure add constraint FKsous_composition
    foreign key (sou_Id)
    references Structure;

alter table Structure add constraint FKconstitution
    foreign key (con_sc_Nom)
    references Scenario;

-- Index Section
-- _____

create unique index IDColonnes
    on Colonnes (Id);

create index FKcolonne
    on Colonnes (col_Id);

create unique index IDcomposition_page
    on composition_page (Id, com_Id);

create index FKest_compose_de
    on composition_page (Id);

```

```

create index FKcompose
  on composition_page (com_Id);

create unique index IDFrame
  on Frame (Id);

create unique index FKsource_Frame
  on Frame (sou_Id);

create index FKdecomposition
  on Frame (F_F_Id);

create unique index IDFrameSet
  on FrameSet (Id);

create unique index FKrepresentation_frameset
  on FrameSet (rep_Id);

create unique index IDLien
  on Lien (Id);

create index FKsource
  on Lien (L_N_Id);

create index FKDestination_structure
  on Lien (L_S_Id);

create index FKDestination_Page
  on Lien (L_P_Id);

create unique index IDNavigation
  on Navigation (Id);

create index FKRepresentation_page
  on Navigation (N_S_Id);

create index FKAppartenance
  on Navigation (App_Id);

create unique index IDPage
  on Page_ (Id);

create index FKconstit_s_P
  on Page_ (con_Nom);

create unique index IDRangées
  on Rangees (Id);

create index FKRangee
  on Rangees (R_F_Id);

```

```
create unique index IDScenario  
on Scenario (Nom);
```

```
create unique index FKDepart  
on Scenario (Id);
```

```
create unique index IDStructure  
on Structure (Id);
```

```
create index FKsous_composition  
on Structure (sou_Id);
```

```
create index FKconstitution  
on Structure (con_sc_Nom);
```